

## **RANCANG BANGUN ALAT PEMILAH BAWANG MERAH BERDASARKAN UKURAN DIAMETER**

Nama : Moh. Roni Hidayat  
NRP : 1109100013  
Pembimbing : Endarko, M.Si, Ph.D.

### **ABSTRAK**

Telah dilakukan perancangan dan pembuatan alat pemilah bawang merah berdasarkan ukuran diameternya. Alat ini memanfaatkan pengolahan citra digital dalam memproses data yang diambil. Data yang diambil dalam penelitian ini berupa data citra berformat *bitmap* yang diambil menggunakan *webcam*. Data citra yang diperoleh kemudian diolah dengan menggunakan *software Visual Studio 2010*. Dari hasil pengolahan citra data yang diperoleh maka akan diperoleh nilai ukuran bawang merah yang kemudian akan dikelompokkan kualitasnya berdasarkan ukuran hasil pengolahan data dari alat. Pengujian alat dilakukan terhadap 51 siung bawang merah dengan kualitas yang berbeda-beda. Dari penelitian ini dapat disimpulkan bahwa telah berhasil difabrikasi sebuah alat untuk memilah bawang merah berdasarkan ukurannya dengan kecepatan 6 siung per menit. Sistem yang dibuat memiliki kemampuan untuk mengenali tiga perbedaan ukuran. Alat yang telah dibuat dalam Tugas Akhir ini telah berhasil memilah 51 bawang merah dalam tiga kategori ukuran yaitu kelas I (20 siung) dengan error 1.25%, kelas II (11 siung) dengan kesalahan 3.06% dan kelas III (20 siung) dengan kesalahan 3.04%. Perbandingan antara hasil pengukuran alat dengan ukuran sebenarnya adalah sebanding dengan persamaan  $y = 0,9977x + 4,2791$ .

Kata kunci : Bawang merah, pengolahan citra, *bitmap*,  
*Visual Studio 2010*, ukuran diameter

**”Halaman ini sengaja dikosongkan”**

## **DESIGN AND FABRICATION OF SHALLOT SORTER DEVICE BASED ON THE SIZE OF DIAMETERS**

Name : Moh. Roni Hidayat  
NRP : 1109100013  
Supervisor : Endarko, M.Si, Ph. D

### **ABSTRACT**

The shallot sorter device based on its diameters has been designed and fabricated. The device used the application of image processing for processing data from the diameter of shallot. The data of a shallot image in the bitmap format have been taken using webcam. All image data have been processed with Visual Studio 2010. The study was to fabricate a shallot sorter device for grading three classes of shallot based on its diameters. The result showed that the system can be sorted the shallot based on the size of diameter with speed at 6 cloves/minutes. Furthermore, the system has been tested to sort the 51 cloves of shallot in three difference of diameters. The result showed that the system has capability to sort in three class with error reading of 1.25% for Class I (20 cloves) and 3.06% for Class II (11 cloves) whilst resulted in 3,04% of error reading for Class III (20 cloves). Comparison between the results of the measurement of the system to the actual size was comparable with the equation  $y = 0.9977x + 4.2791$ .

Keyword : Shallot, image processing, *bitmap*, *Visual Studio 2010*, diameter

**”Halaman ini sengaja dikosongkan”**

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Bawang Merah**

Tanaman bawang merah berasal dari Syiria, entah beberapa ribu tahun yang lalu sudah dikenal umat manusia sebagai penyedap masakan (Rismunandar, 1986). Sekitar abad VII tanaman bawang merah mulai menyebar ke wilayah Eropa Barat, Eropa Timur dan Spanyol, kemudian menyebar luas ke daratan Amerika, Asia Timur dan Asia Tenggara (Singgih 1991). Pada abad XIX bawang merah telah menjadi salah satu tanaman komersial di berbagai negara di dunia. Negara-negara produsen bawang merah antara lain adalah Jepang, USA, Rumania, Italia, Meksiko dan Texas (Rahmat, 1994).

Di Indonesia, daerah yang merupakan sentra produksi bawang merah adalah Cirebon, Brebes Tegal, Kuningan, Wates (Yogyakarta), Lombok Timur dan Samosir (Sunarjono dan Soedomo, 1989). Pada tahun 2003, total pertanaman bawang merah petani Indonesia sekitar 88,029 hektar dengan rata-rata hasil 7,6 ton/ha (Putrasamedja, Sartono 2010). Produktivitas hasil bawang merah tersebut dipandang masih rendah, karena potensi hasil yang dapat dicapai sekitar 20 ton/ha.

Bawang merah dapat dipanen setelah umurnya cukup tua, biasanya pada umur 60 – 70 hari. Tanaman bawang merah dipanen setelah terlihat tanda-tanda 60% leher batang lunak, tanaman rebah, dan daun menguning. Pemanenan sebaiknya dilaksanakan pada keadaan tanah kering dan cuaca yang cerah untuk mencegah serangan penyakit busuk umbi di gudang. Bawang merah yang telah dipanen kemudian diikat pada batangnya untuk mempermudah penanganan. Selanjutnya umbi dijemur sampai cukup kering dengan di bawah sinar matahari langsung, kemudian biasanya diikuti dengan pengelompokan berdasarkan kualitas umbi. Apabila tidak

langsung dijual, umbi bawang merah disimpan dengan cara menggantungkan ikatan-ikatan bawang merah di gudang khusus, pada suhu 25 – 30 °C dan kelembaban yang cukup rendah yaitu  $\pm$  60 – 80% (Sutarya dan Grubben, 1995).

### **2.1.1 Standar Bawang Merah**

Standar bawang merah disusun berdasarkan survai didaerah penghasil bawang merah yaitu di Jawa Tengah dan Jawa Timur, studi pustaka serta wawancara dengan Dinas Pertanian setempat dan Lembaga Penelitian Hortikultura.

Setelah mempelajari hasil survai tersebut serta memperbandingkannya dengan standar mutu “onibas” dari Amerika Serikat maka disusunlah Standar Bawang Merah Indonesia sebagai berikut (Badan Standarisasi Nasional, 1992):

#### **Spesifikasi**

1. Ruang Lingkup.  
Standar ini meliputi syarat mutu, cara pengujian mutu, cara pengambilan contoh dan pengamanan bawang merah.
2. Deskripsi.  
Bawang merah adalah umbi lapis tanaman bawang merah (*Allium Ascalonicum L*) yang terdiri dari siung-siung bernas, utuh, segar dan bersih.
3. Jenis Mutu.  
Bawang merah digolongkan dalam dua jenis mutu, yaitu Mutu I dan Mutu II.
4. Syarat Mutu.

Tabel 2.1 Syarat mutu bawang merah

Karakteristik	Syarat		Cara Pengujian
	Mutu 1	Mutu 2	
Kesamaan sifat varietas	seragam	seragam	Organoleptik
Ketuaan	Tua	Cukup tua	Organoleptik
Kekerasan	Keras	Cukup keras kompak	Organoleptik
Diameter (cm) min.	1,7	1,3	SP-SMP-309-1981
Kerusakan, % (bobot) maks	5	8	SP-SMP-309-1981
Busuk, % (bobot) maks	1	2	SP-SMP-309-1981
Kotoran, % (bobot) maks	Tidak ada	Tidak ada	SP-SMP-309-1981

## Keterangan:

- Kesamaan sifat varietas : Kesamaan sifat varietas dinyatakan seragam apabila bawang merah dalam satu slot seragam dalam bentuk umum umbi.
- Ketuaan : Bawang merah dinyatakan tua, apabila bawang merah telah mencapai tingkat pertumbuhan fisiologis yang cukup tua, dimana umbinya cukup padat dan tidak lunak.
- Kekerasan : Bawang merah dinyatakan keras, apabila umbi bawang merah setelah mengalami “curing” / pengeringan dengan baik cukup keras dan tidak lunak bila ditekan dengan jari.
- Diameter : adalah dimensi terbesar diukur tegak lurus pada garis lurus sepanjang batang sampai akar.
- Kerusakan : Bawang merah dinyatakan rusak apabila mengalami kerusakan atau cacat oleh sebab fisiologis, mekanis dan lain-lain yang terlihat pada permukaan.

- Busuk : Bawang merah dinyatakan busuk apabila mengalami pembusukan akibat kerusakan biologis.
- Kotoran : Yang dimaksud dengan kotoran adalah semua bahan bukan bawang merah atau benda asing lainnya (seperti tanah bahan tanaman dan lain-lain) yang menempel atau berada dalam kemasan, yang mempengaruhi kenampakannya, bahan penyekat / pembungkus tidak dianggap sebagai kotoran.

## 2.2. Pengolahan Citra Digital

Citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimesi. Secara matematis fungsi intensitas cahaya pada bidang dua dimensi disimbolkan dengan  $f(x, y)$ , yang dalam hal ini (Rafael,1977)

$(x, y)$  : koordinat pada bidang dua dimensi

$f(x, y)$  : intensitas cahaya (*brightness*) pada titik  $(x, y)$

Agar dapat diolah dengan dengan komputer, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit. Representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut *digitalisasi* . Citra yang dihasilkan inilah yang disebut citra digital (*digital image*). Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi ukurannya dinyatakan sebagai tinggi x lebar (atau lebar x panjang) (Hans,1997).

### 2.2.1. Elemen Dasar Citra Digital

Citra digital mengandung sejumlah elemen-elemen dasar. Elemen-elemen dasar tersebut dimanipulasi dalam pengolahan citra dan dieksploitasi lebih lanjut dalam komputer. Elemen-elemen dasar yang penting diantaranya adalah (Mengko, 1989):



- Kecerahan (*brightness*).

Kecerahan merupakan intensitas cahaya namun kecerahan pada sebuah titik (*pixel*) di dalam citra bukanlah intensitas yang riil, melainkan adalah intensitas rata-rata dari suatu area yang melingkupinya. Sistem visual manusia mampu menyesuaikan dirinya dengan tingkat kecerahan (*brightness level*) mulai dari yang paling rendah sampai yang paling tinggi dengan jangkauan sebesar  $10^{10}$  (Mengko, 1989).

- Kontras (*contrast*).

Kontras menyatakan sebaran terang (*lightness*) dan gelap (*darkness*) di dalam sebuah gambar. Citra dengan kontras rendah dicirikan oleh sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

- Kontur (*contour*)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada *pixelpixel* yang bertetangga. Karena adanya perubahan intensitas inilah mata kita mampu mendeteksi tepi-tepi (*edge*) objek di dalam citra.

- Warna (*color*)

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang ( $\lambda$ ) yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu (*violet*) mempunyai panjang gelombang paling rendah. Warna-warna yang diterima oleh mata (sistem visual manusia) merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red* ( $R$ ), *green* ( $G$ ), dan *blue* ( $B$ ). Persepsi sistem visual manusia terhadap warna sangat relatif sebab dipengaruhi oleh banyak kriteria, salah satunya disebabkan oleh adaptasi yang menimbulkan distorsi. Misalnya bercak abu-abu di sekitar warna hijau akan tampak keunguan (distorsi terhadap ruang), atau jika mata melihat warna hijau lalu langsung dengan

cepat melihat warna abu-abu, maka mata menangkap kesan warna abu-abu tersebut sebagai warna ungu (distorsi terhadap waktu) (Mengko, 1989).

- Bentuk (*shape*)

*Shape* adalah properti intrinsik dari objek tiga dimensi, dengan pengertian bahwa *shape* merupakan properti intrinsik utama untuk sistem visual manusia. Manusia lebih sering mengasosiasikan objek dengan bentuknya ketimbang elemen lainnya (warna misalnya). Pada umumnya, citra yang dibentuk oleh mata merupakan citra dua dimensi), sedangkan objek yang dilihat umumnya berbentuk tiga dimensi). Informasi bentuk objek dapat diekstraksi dari citra pada permulaan pra-pengolahan dan segmentasi citra. Salah satu tantangan utama pada *computer vision* adalah merepresentasikan bentuk, atau aspek-aspek penting dari bentuk (Ballard, 1982).

- Tekstur (*texture*)

Tekstur dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan *pixel-pixel* yang bertetangga. Sistem visual manusia pada hakikatnya tidak menerima informasi citra secara independen pada setiap *pixel*, melainkan suatu citra dianggap sebagai suatu kesatuan. Resolusi citra yang diamati ditentukan oleh skala pada mana tekstur tersebut dipersepsi (Jain, 1995).

### 2.2.2 Operasi Dasar Pengolahan Citra Digital

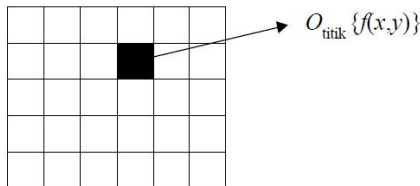
Citra digital diolah dengan menggunakan komputer, oleh karena itu penting untuk mendefinisikan struktur data untuk merepresentasikan citra di dalam memori komputer. Matriks adalah struktur data yang tepat untuk merepresentasikan citra digital. Elemen-elemen matriks dapat diakses secara langsung melalui indeksnya (baris dan kolom). Operasi pada citra digital pada dasarnya adalah memanipulasi elemen-elemen matriks. Elemen matriks yang dimanipulasi dapat berupa elemen tunggal (sebuah *pixel*), sekumpulan elemen yang berdekatan, atau

keseluruhan elemen matriks. Operasi-operasi yang dilakukan pada pengolahan citra dapat dikelompokkan ke dalam empat aras (*level*) komputasi, yaitu (Jain, 1995) :

- Aras Titik

Operasi pada aras titik hanya dilakukan pada *pixel* tunggal di dalam citra. Operasi titik dikenal juga dengan nama operasi *pointwise*. Operasi ini terdiri dari pengaksesan *pixel* pada lokasi yang diberikan, memodifikasinya dengan operasi linjar (*linear*) atau nirlinjar (*nonlinear*), dan menempatkan nilai *pixel* baru pada lokasi yang bersesuaian di dalam citra yang baru. Operasi ini diulangi untuk keseluruhan *pixel* di dalam citra. Secara matematis, operasi pada aras titik dinyatakan sebagai (Gambar 2.1):

$$fB(x, y) = O_{\text{titik}}\{fA(x, y)\} \quad (2.1)$$



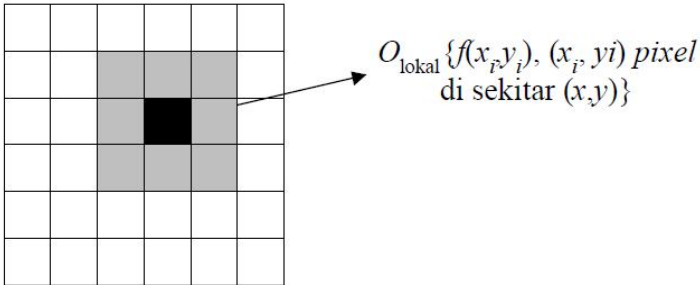
Gambar 2.1 Operasi Aras Titik pada Citra Digital

yang dalam hal ini  $fA$  dan  $fB$  masing-masing adalah citra masukan dan citra keluaran,  $O_{\text{titik}}$  dapat berupa operasi linjar (*linear*) atau nirlinjar (*nonlinear*). Yang dimaksud dengan operasi linjar adalah operasi yang dapat dinyatakan secara matematis sebagai persamaan linjar, sebaliknya adalah persamaan nirlinjar. Operasi pada aras titik dapat dibagi menjadi tiga macam: berdasarkan intensitas, berdasarkan geometri, atau gabungan keduanya (Jain, 1995).

- Aras Lokal

Operasi pada aras lokal menghasilkan citra keluaran yang intensitas suatu *pixel* bergantung pada intensitas *pixel-pixel* tetangganya (Gambar 2.2).

$$fB(x, y)' = O_{\text{lokal}}\{fA(x_i, y_j); (x_i, y_j) \in N(x, y)\} \quad (2.2)$$



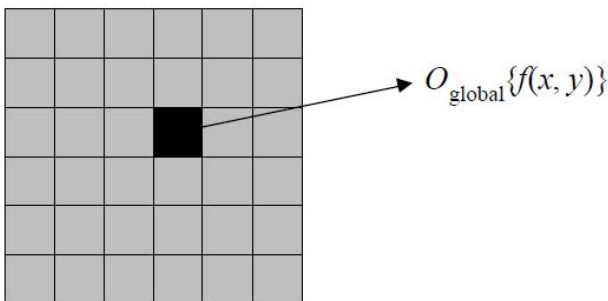
Gambar 2.2 Operasi Aras Lokal

dimana  $N = \text{neighborhood}$ , yaitu *pixel-pixel* yang berada di sekitar  $(x, y)$ . Contoh operasi beraras lokal adalah operasi konvolusi untuk mendeteksi tepi (*edge detection*) dan pelembutan citra (*image smoothing*).

- Aras Global

Operasi pada aras global menghasilkan citra keluaran yang intensitas suatu *pixel* bergantung pada intensitas keseluruhan *pixel* (Gambar 2.3).

$$fB(x, y)' = O_{\text{global}}\{fA(x, y)\} \quad (2.3)$$



Gambar 2.3 Operasi Aras Global

Contoh operasi beraras global adalah operasi penyetaraan histogram untuk meningkatkan kualitas citra.

- Aras Objek

Operasi jenis ini hanya dilakukan pada objek tertentu di dalam citra. Tujuan dari operasi pada aras objek adalah untuk mengenali objek tersebut, misalnya dengan menghitung rata-rata intensitas, ukuran, bentuk, dan karakteristik lain dari objek. Operasi aras objek adalah operasi yang sangat sulit, karena sebelumnya kita harus dapat menjawab: apakah objek itu, bagaimana menemukannya.

### 2.2.3 Operasi dalam Citra Digital

Karena citra digital adalah matriks, maka operasi-operasi aritmetika matriks juga berlaku pada citra. Operasi matriks yang dapat dilakukan adalah:

- Penjumlahan atau pengurangan antara dua buah citra  $A$  dan  $B$

$$C(x, y) = A(x, y) \pm B(x, y), \quad (2.4)$$

Operasi penjumlahan citra dapat digunakan untuk mengurangi pengaruh derau (*noise*) di dalam data, dengan cara merata-ratakan derajat keabuan setiap *pixel* dari citra yang sama yang diambil berkali-kali. Misalnya untuk citra yang sama direkam dua kali,  $f_1$  dan  $f_2$ , lalu dihitung intensitas rata-rata untuk setiap *pixel*:

$$f'(x, y) = 0,5 \{ f_1(x, y) + f_2(x, y) \} \quad (2.5)$$

Hasil operasi mungkin bernilai riil, karena itu semua nilai riil tersebut perlu dibulatkan ke nilai bulat terdekat, nilai maksimum adalah 255.

$C$  pada persamaan (2.4) adalah citra baru yang intensitas setiap *pixel*-nya adalah selisih antara intensitas *pixel* pada  $A$  dan

*B*. Ada kemungkinan hasil operasi ini menghasilkan nilai negatif, oleh karena itu, operasi pengurangan citra perlu melibatkan operasi *clipping*. Contoh aplikasi operasi pengurangan citra adalah untuk memperoleh suatu objek dari dua buah citra. Pengurangan citra juga dapat digunakan untuk mendeteksi perubahan yang terjadi selama selang waktu tertentu bila dua buah citra yang diambil adalah citra dari adegan yang sama. Teknik semacam ini dipakai pada *moving images*.

(Hendradjaya, 1995).

- Perkalian dua buah citra

$$C(x, y) = A(x, y) B(x, y) \quad (2.6)$$

Perkalian citra sering digunakan untuk mengoreksi kenirlanjaran sensor dengan cara mengalikan matriks citra dengan matrik koreksi. Jadi, dalam hal ini *A* adalah citra sedangkan *B* adalah matriks koreksi. Hasil operasi mungkin bernilai riil, karena itu semua nilai dibulatkan ke nilai bulat terdekat, nilai maksimum adalah 255.

- Penjumlahan/pengurangan citra *A* dengan skalar *c*:

$$B(x, y) = A(x, y) \pm c, \quad (2.7)$$

Penjumlahan citra *A* dengan skalar *c* adalah menambah setiap *pixel* di dalam citra dengan sebuah skalar *c*, dan menghasilkan citra baru *B* yang intensitasnya lebih terang daripada *A*. Kenaikan intensitas sama untuk seluruh *pixel*, yaitu *c*. Pengurangan citra *A* dengan skalar *c* adalah mengurangi setiap *pixel* di dalam citra dengan sebuah skalar *c*, dan menghasilkan citra baru *B* yang intensitasnya lebih gelap daripada *A*. Penurunan intensitas sama untuk seluruh *pixel*, yaitu *c*. Contoh operasi penjumlahan/pengurangan citra dengan sebuah skalar adalah operasi pencerahan citra (lihat pembahasan operasi aras titik). Baik operasi penjumlahan maupun pengurangan citra dengan sebuah skalar melibatkan operasi *clipping*.

- Perkalian/pembagian citra  $A$  dengan sebuah skalar  $c$ :

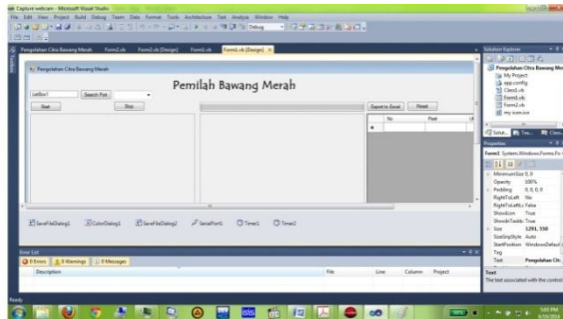
$$B(x, y) = c \times A(x, y) \quad (2.8)$$

Perkalian citra  $A$  dengan skalar  $c$  menghasilkan citra baru  $B$  yang intensitasnya lebih terang daripada  $A$ . Kenaikan intensitas setiap *pixel* sebanding dengan  $c$ . Operasi perkalian citra dengan skalar dipakai untuk kalibrasi kecerahan (*callibration of brightness*). Pembagian citra  $A$  dengan skalar  $c$  menghasilkan citra baru  $B$  yang intensitasnya lebih gelap daripada  $A$ . Penurunan intensitas setiap *pixel* berbanding terbalik dengan  $c$ . Operasi pembagian citra dengan skalar dipakai untuk normalisasi kecerahan (*normalization of brightness*).

Selain terdapat operasi aritmatika, dalam pengolahan citra digital juga terdapat operasi Boolean (and, or, dan not) dan juga operasi geometri (translasi, rotasi, penskalaan dan *flipping*). Operasi Boolean mempunyai terapan yang penting pada pemrosesan morfologi pada citra biner. Pada citra biner, operasi not dapat digunakan untuk menentukan komplemen dari citra. Sedangkan pada operasi geometrik, koordinat *pixel* berubah akibat transformasi, namun intensitasnya tetap. Ini berbeda dengan dengan operasi aritmetika dimana koordinat *pixel* tetap sedangkan intensitasnya berubah.

## 2.3 Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 Ultimate merupakan suatu IDE program yang menyediakan berbagai bahasa terintegrasi dengan komponen-komponen pendukung lainnya untuk pembuatan aplikasi dilingkungan Microsoft. Tidak hanya berbasis dekstop, Visual Studio juga memiliki untuk kemampuan untuk membuat aplikasi berbasis web yaitu dengan ASP, yang dilengkapi dengan MSSQL Server sebagai database defaultnya.



Gambar 2.4. Tampilan Microsoft Visual Studio 2010

Mengenal tampilan visual basic 2010 yang terdapat beberapa bagian yaitu:

a. Tittle bar

Tittle Bar adalah tempat untuk menampilkan nama project yang sedang dibuat.

b. Menu bar

Menu bar yang terdapat pada program– program aplikasi di Windows. Menu Bar digunakan untuk melakukan proses atau perintah- perintah tertentu. Menu bar dibagi menjadi beberapa pilihan sesuai dengan kegunaan nya, seperti menu bar File digunakan untuk memproses atau menjalankan perintah-perintah yang berhubungan dengan file, seperti membuka file baru, menyimpan file, selain itu juga terdapat Menu Bar lain seperti : Edit, View, Project, Build, Debug, Data, Format, Tools, Window, dan Help. Untuk menggunakan Menu Bar, dapat dilakukan dengan dua cara, yaitu :

1. Dengan mouse, klik mouse pada menu dan sub menu.
2. Dengan Keyboard, tekan ALT dan karakter bergaris bawah (Foxall, 2010).



### c. Toolbars

Toolbars pada aplikasi windows lainnya yang berisi tombol-tombol yang mewakili suatu perintah tertentu yang sering digunakan untuk keperluan dalam pemrograman dan lain-lain, toolbars dapat kita lihat dalam bentuk icon.

### d. Solution Explorer

Solution Explorer adalah jendela yang menyimpan Informasi mengenai Solution, Project – project, beserta file-file, form – form ataupun resource yang digunakan pada program aplikasi. Pada bagian atas jendela Solution Explorer terdapat toolbox yang digunakan untuk menampilkan jendela Properties, menampilkan semua file, melihat Design form, Refresh dan View code, untuk melihat kode program. Pada Solution explorer juga kita dapat menambahkan class, module, windows form baru, dan sebagainya. Jika pada saat kita mengaktifkan Visual Studio 2010 jendela Solution Explorer tidak ada, kita dapat menampilkannya dari menu bar View, Solution Explorer atau menggunakan tombol Ctrl + Alt + L.

### e. Form

Form Designer merupakan suatu objek yang digunakan untuk merancang tampilan program. Form Designer juga dapat dikatakan sebagai objek utama pada pemrograman Visual Basic karena pada form inilah nantinya Komponen dan kontrol Toolbox diletakan dan diatur sebagai mungkin. Form dapat diatur melalui jendela Properties. Ukuran Form Designer ini juga dapat diubah tinggi dan lebarnya, dengan cara mengklik pada Form Designer tersebut, sehingga tampak garis putus-putus disekelilingnya, hanya dengan men-Drag Form ke kiri, kanan, atas, ataupun bawah, maka ukuran Form akan berubah. (Willis, 2010).

#### f. Toolbox

ToolBox Standar yang terdapat pada Visual Basic 2010 adalah tempat penyimpanan kontrol-kontrol atau komponen standar yang nantinya akan kita letakkan sebagai komponen program didalam Form saat merancang sebuah aplikasi. ToolBox adalah tempat dimana kontrol dan komponen yang dilambangkan dengan icon. Kontrol dan komponen sangat membantu pada saat proses merancang tampilan Form dalam pembuatan program. kontrol dan komponen diletakkan pada tab-tab berdasarkan kegunaannya. Apabila saat kita menjalankan Visual Studio 2010, Jendela toolbox tidak ada, maka kita dapat menampilkan nya melalui menu bar View > ToolBox atau dengan menggunakan gabungan tombol Ctrl + Alt X. Pada jendela toolbox kita dapat mengaktifkan tab yang akan ditampilkan dengan cara mengklik tanda “+” pada sisi kiri tab toolbox.

#### g. Properties

Jendela Properties berfungsi untuk memberikan informasi mengenai objek yang sedang aktif, nama objek yang sedang aktif dapat dilihat pada bagian atas jendela Properties. Properties juga digunakan untuk merubah nilai property atau karakteristik dari objek yang aktif. Komponen-komponen atau kontrol-kontrol VB 2010 mempunyai property dan event yang berbeda untuk satu dan lainnya, tetapi ada juga yang memiliki property dan event yang sama. Pada VB 2008 jendela properties terbagi atas bagian / jenis, pertama yaitu properties yang berfungsi untuk menampung property masing-masing objek serta pada bagian ini juga karakteristik dari komponen tersebut dapat diatur atau dirubah. Property merupakan setiap komponen di dalam pemrograman Visual Basic 2010 dapat diatur propertinya sesuai dengan kebutuhan aplikasi. Property yang tidak boleh dilupakan pada setiap komponen adalah “Name”, yang berarti nama variabel (komponen) yang akan digunakan dalam scripting. Properti “Name” ini hanya bisa diatur melalui jendela Property.

#### h. Error list

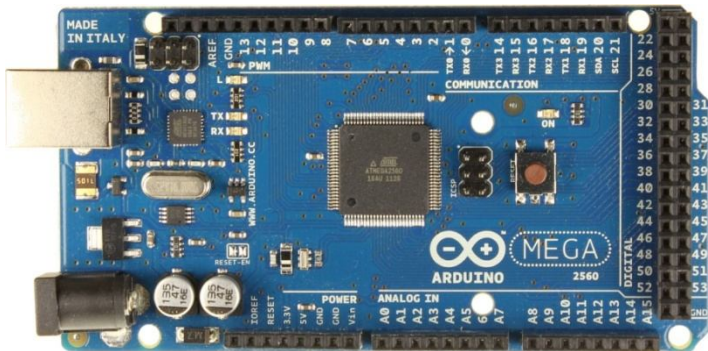
Pada bagian ini biasanya berisi tentang error dan warning. Contoh error ialah penulisan kode yang belum di deklarasikan. Sedangkan untuk warning berisi himbauan semisal ada kode yang seharusnya dapat tidak ditulis karena tidak dipakai (Novák, 2010).

## 2.4 Arduino

Arduino merupakan modul *single board* berbasis mikrokontroler yang berifat *open source* (baik *hardware* maupun *software*), diturunkan dari Wiring Platform, dan dirancang untuk memudahkan *user* dalam penggunaan elektronik dalam berbagai bidang. Arduino dapat dikatakan sebagai sebuah *platform* dari *physical computing* yang bersifat *open source*. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi merupakan kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment* (IDE). IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam memory mikrokontroler.

### 2.4.1 Hardware Arduino

Aduino Hardware hingga saat ini berbasis mikrokontroler AVR 8 bit RISC (seri ATmega) seperti ATmega168, ATmega328, ATmega1280 dan ATmega2560 yang telah dilengkapi *bootloader* untuk membantu proses pengisian program. Salah satu contoh *hardware* Arduino adalah “Arduino mega 2560” seperti pada Gambar 2.5 di bawah ini:



Gambar 2.5 Arduino Board

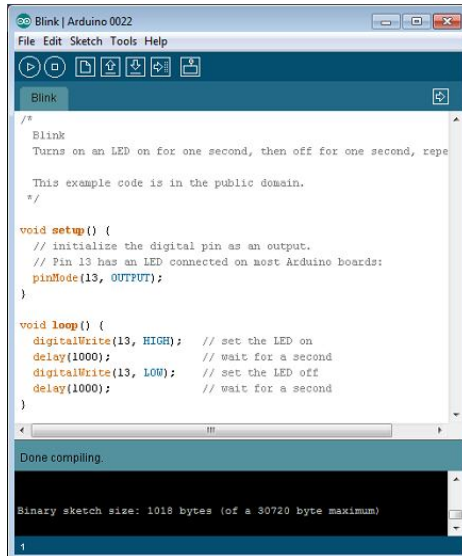
### 2.4.2 Software Arduino

*Software* untuk Arduino biasa disebut dengan IDE Arduino, yakni sebuah *software* yang dikembangkan dengan basis IDE *Processing* yang menggunakan bahasa yang mirip dengan C++ dan Java. Bahasa dan alur pemrogramannya cukup mudah dimengerti walaupun cukup “*Case Sensitive*” dan belum dilengkapi dengan *Code Completion* (*AutoComplete*). Arduino Alpha dapat dijalankan pada OS multi *platform* seperti Windows, Linux, dan Macintosh. Arduino Alpha bersifat *Executable File* dan dapat langsung dijalankan tanpa harus melakukan proses instalasi terlebih dahulu asalkan pada komputer telah terinstalasi Java Runtime. IDE Arduino Alpha terdiri dari:

- a. **Editor program**, sebuah *window* yang memungkinkan *user* menulis dan melakukan proses *editing* program dalam bahasa *Processing*.
- b. **Compiler**, sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner.

- c. **Uploader**, sebuah modul yang berfungsi memuat (meng-*upload*) kode biner dari komputer ke dalam memory di dalam Arduino board.

Berikut ini adalah contoh tampilan program Arduino IDE:



Gambar 2.6 Program Arduino IDE (Arduino, 2014)

## 2.5 Motor Stepper

Motor Stepper adalah motor DC yang gerakannya bertahap (*step per step*) dan memiliki akurasi yang tinggi tergantung pada spesifikasinya. Setiap motor stepper mampu berputar untuk setiap stepnya dalam satuan sudut (0.75, 0.9, 1.8), makin kecil sudut per step-nya maka gerakan per step-nya motor stepper tersebut makin presisi (anonim, 2007).

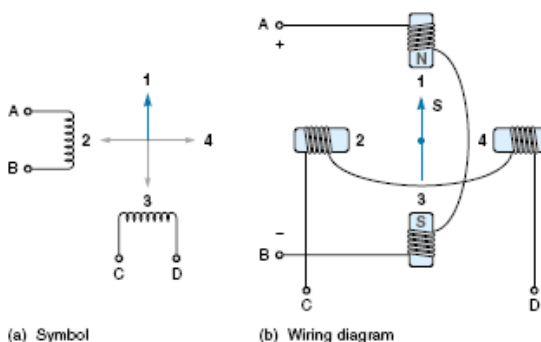
### 2.5.1 Motor Stepper Dua-phase (*Bipolar*)

Motor stepper dua-phase (*bipolar*) mempunyai konstruksi yang mirip dengan jenis unipolar, hanya tidak terdapat *tap* pada kumparannya (gambar 7). Penggunaan motor stepper jenis bipolar memerlukan rangkain yang agak lebih rumit untuk mengatur agar motor ini dapat berputar dalam dua arah. Untuk menggerakkan motor stepper jenis ini biasanya diperlukan sebuah *driver* motor yang dikenal dengan nama *H bridge*. Rangkaian ini akan mengontrol setiap kumparan secara terpisah (*independent*) termasuk polaritas untuk setiap kumparan.

Motor stepper dua-phase (*bipolar*) hanya mempunyai dua rangkaian tetapi sebenarnya terdiri dari empat kutub medan. Gambar 2.7(a) menunjukkan simbol motor dan Gambar 2.7(b) menunjukkan bagaimana perkawatan/lilitan internal motor tersebut. Pada Gambar 2.7(b), rangkaian AB terdiri dari dua kutub berlawanan sedemikian bila tegangan yang dikenakan (+A-B), kutub bagian atas akan memberikan ujung utara terhadap rotor dan kutub bawah akan memberikan ujung selatan. Rotor akan cenderung sejajar sendiri secara vertikal (posisi 1) dengan kutub selatannya mengarah ke atas (sebab kutub magnet yang berlawanan akan saling menarik).

Cara yang paling sederhana dalam memberikan *step* pada motor ini adalah dengan memberikan energi secara bergantian pada AB atau CD untuk menarik rotor dari kutub ke kutub. Jika rotor bergerak CCW (*counterclockwise*, berlawanan arah jarum jam) dari posisi 1, maka rangkaian CD harus diberi energi dengan polaritas C+D-. Hal ini akan menarik rotor ke posisi 2. Selanjutnya, rangkaian AB diberi energi lagi, tetapi kali ini polaritasnya terbalik (-A+B), yang menyebabkan kutub bawah memberikan ujung utara pada rotor, dengan demikian tertarik ke posisi 3. Istilah bipolar digunakan pada motor ini karena arus kadang-kadang terbalik. Urutan tegangan diperlukan untuk

memutar motor satu putaran penuh dan ditunjukkan di bawah ini. Pembacaan dari atas ke bawah memberikan urutan untuk peralihan/perputaran CCW, pembacaan dari bawah ke atas adalah urutan CW (*clockwise*, sarah jarum jam):



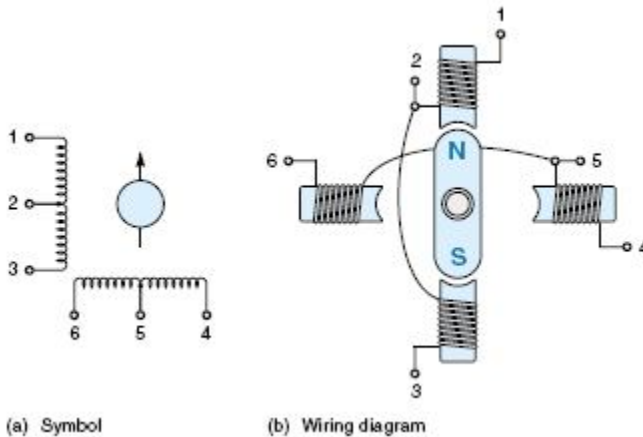
Gambar 2.7 Motor stepper dua-phase (*bipolar*)  
(Syahrul, 2011)

### 2.5.2 Motor Stepper Empat-Phase (*Unipolar*)

Motor stepper empat-phase (unipolar) adalah jenis motor stepper yang paling umum (Gambar 2.8). Istilah empat-phase digunakan karena motor mempunyai empat kumparan medan yang dapat diberikan energi secara terpisah/tersendiri, dan istilah unipolar digunakan karena arus selalu menjalar dalam arah yang sama melalui kumparan. Cara sederhana untuk mengoperasikan motor stepper empat-phase adalah dengan memberikan energi phase satu pada suatu waktu yang berurutan (dikenal dengan *wave drive*). Untuk memutar ke arah CW, digunakan urutan berikut:

Dibandingkan dengan motor stepper bipolar dua-phase, motor stepper empat-phase mempunyai keuntungan karena kesederhanaannya (*simplicity*). Rangkaian kontrol motor empat-phase mudah men-*switch* urutan kutub on dan off ; tanpa harus

membalik polaritas kumparan medan (namun, motor dua-phase menghasilkan torsi lebih besar karena *pushing* dan *pulling* dilakukan bersamaan).



Gambar 2.8 Motor stepper empat-phase dengan kumparan *center tap* (Syahrul, 2011).

Dengan konstruksi motor demikian mereka dapat digunakan dalam mode dua-phase atau empat-phase seperti yang biasa digunakan. Hal ini dilakukan dengan membiarkan dua kumparan tambahan (dari motor dua-phase) yang secara internal dihubungkan ke titik di antara kumparan medan yang berlawanan. Gambar 2.8(a) menunjukkan simbol untuk jenis motor ini dan Gambar 2.8(b) menunjukkan interior kumparan motor. Jika motor tersebut digunakan dalam mode dua-phase, maka *center tap* (terminal 2 dan 5) tidak digunakan. Jika dioperasikan dalam mode empat-phase, *center tap* menjadi *common return*, dan *power* diberikan pada terminal 1, 4, 3 dan 6 seperti yang dibutuhkan (Syahrul, 2011).



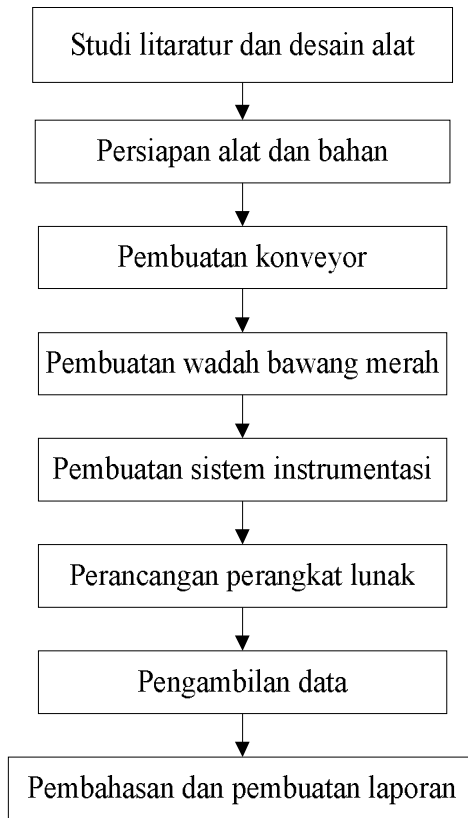
### **BAB III**

## **METODOLOGI PENELITIAN**

Hal yang akan dibahas pada bab ini adalah mengenai perancangan dan pembuatan alat. Dalam perancangan pembuatan alat ini diawali dengan perancangan umum sistem dari keseluruhan (Gambar 3.2). Adapun perancangan pembuatan rancang bangun sistem ini terbagi atas beberapa perangkat yang saling berhubungan yaitu perangkat elektronik dan non elektronik serta perangkat lunak yang berisi instruksi untuk menjalankan program.

### **3.1 Diagram Alir Penelitian**

Secara garis besar, penelitian ini dilakukan sebagaimana diagram alir berikut ini (Gambar 3.1):



Gambar 3.1 Diagram alir penelitian

### 3.2 Alat dan Bahan

Peralatan yang digunakan dalam penelitian ini terdiri atas bawang merah, webcam, laptop, konveyor, wadah bawang merah, arduino mega 2560, rangkaian catu daya, serta seperangkat rangkaian driver motor stepper.

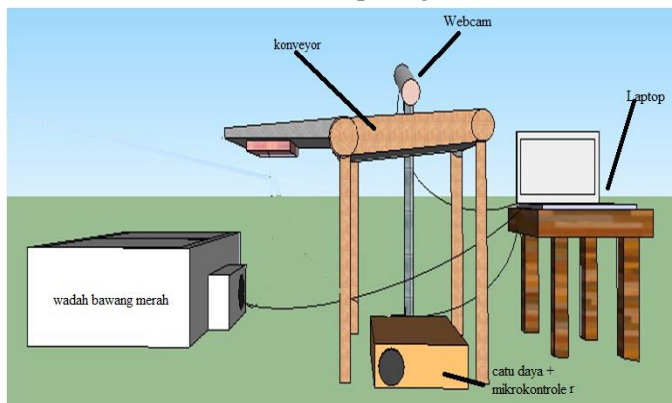
### 3.3 Langkah Kerja

#### 3.3.1 Studi Literatur

Studi literatur digunakan untuk mengetahui konsep dasar yang berkaitan dengan penelitian tugas akhir dan cara kerja peralatan-peralatan yang digunakan. Literatur yang digunakan dalam tugas akhir ini meliputi buku teks, artikel dan jurnal ilmiah serta materi dari internet.

#### 3.3.2 Desain Alat

Secara umum Desain alat seperti gambar 3.2



Gambar 3.2 Desain Pemilah bawang merah

#### 3.3.3 Pembuatan Konveyor

Konveyor ini memiliki ukuran panjang 40 cm, lebar 10 cm dan tinggi 15 cm. Penggeraknya adalah motor stepper. Bahan yang dipilah (bawang merah), nantinya diletakkan di belt konveyor ini dan diatur posisinya.

### 3.3.4 Pembuatan Wadah Bawang Merah

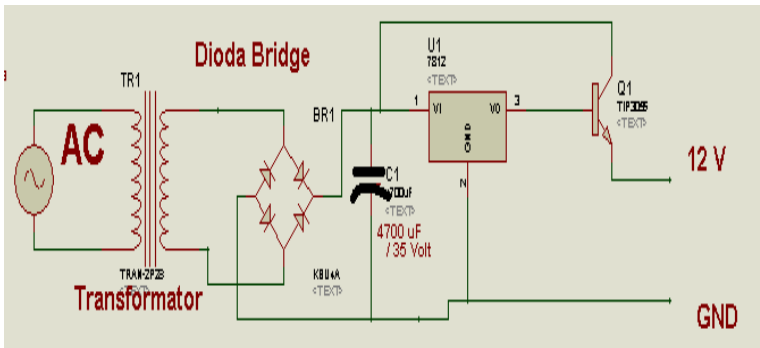
Wadah Bawang merah memiliki 3 buah tempat bawang merah yang masing-masing merupakan berukuran  $10 \times 10 \text{ cm}^2$ . Setiap tempat menyatakan kualitas dari bawang merah, yakni kualitas 1, kualitas 2 dan kualitas 3.

### 3.3.5 Pembuatan Sistem Instrumentasi

#### 3.3.5.1 Catu Daya

Catu daya merupakan sumber tenaga yang dibutuhkan suatu rangkaian elektronika untuk bekerja. Besarnya suplai daya tergantung spesifikasi alat masing-masing. Pada sistem pemilah ini, catu daya digunakan untuk driver motor stepper pada rangkaian.

Rangkaian membutuhkan suplai daya sebesar +12V. Gambar 3.2 merupakan rangkaian catu daya yang dapat memenuhi rangkaian tersebut. inputan rangkaian ini adalah tegangan 220 V, dengan menggunakan LM 7812 dan diberi Transistor Tip 3055. Rangkaian ini menghasilkan keluaran +12 V 5A.



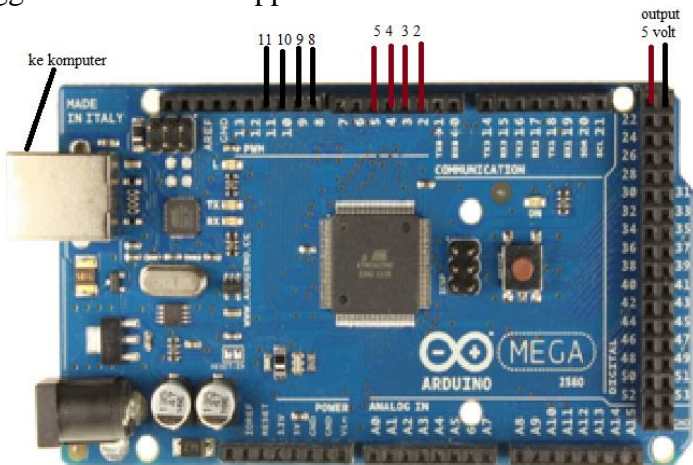
Gambar 3.3 Rangkaian Catu daya

### 3.3.5.2 Arduino Mega 2560

Arduino mega 2560 digunakan untuk mengendalikan motor stepper pada konveyor dan wadah bawang merah. Pada arduino mega 2560, penggerak motor telah khusus disediakan, yakni pada pin 2 sampai pin 13.

Untuk motor stepper 1 pin yang digunakan adalah 2, 3, 4 dan 5 serta satu buah output arduino 5 volt. Sedangkan untuk motor stepper 2 pin yang digunakan adalah 8, 9, 10 dan 11 serta satu buah output arduino 5 volt (lihat Gambar 3.4).

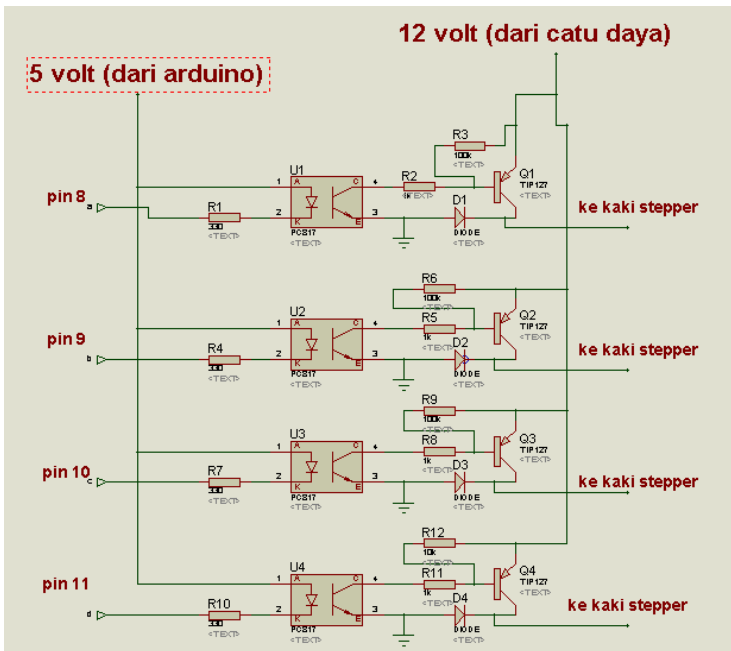
Lalu agar dapat berfungsi dihubungkan ke laptop dengan kabel serial. Selain berfungsi untuk sumber tenaga, kabel serial juga digunakan untuk menerima data dari laptop. Data tersebut nantinya digunakan untuk menggerakkan motor stepper.



Gambar 3.4 letak pin Arduino yang digunakan

### 3.3.5.3 Driver Motor Stepper

Bahan yang digunakan untuk driver motor stepper adalah resistor (330 ohm, 1k ohm, 100 k ohm), optocoupler (tlp 521) transistor TIP 127 serta diode 2 A yang masing-masing komponen 4 buah. Rangkaian ini diberi sumber tegangan 5 volt untuk optocoupler yang terhubung dengan kaki arduino. Sedangkan untuk yang optocoupler yang terhubung dengan transistor dan kaki motor stepper diberi sumber tegangan 12 volt.

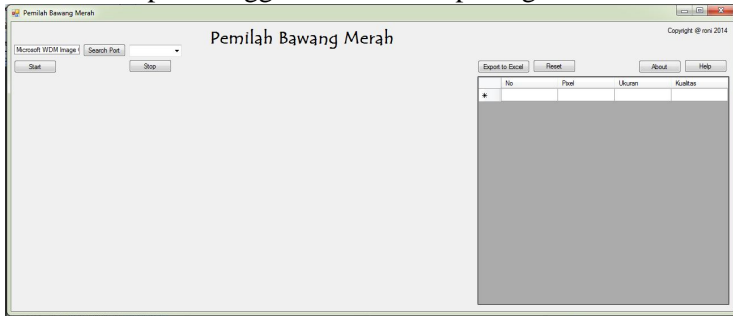


Gambar 3.5 Driver untuk Stepper

### 3.3.6 Perancangan Perangkat Lunak

#### 3.3.6.1 Perancangan Program di Visual Studio

Rancangan Program di Visual Studio 2010 dibuat dengan sedemikian rupa sehingga terbentuk tampilan gambar 3.6.



Gambar 3.6 Tampilan Program Pemilah Bawang Merah

Secara garis besar program tersebut pengambilan citra oleh webcam dan pengolahan citra bawang merah. Hasil dari pengolahan citra tersebut dikirim ke arduino untuk diproses lebih lanjut.

#### 3.3.6.2 Perancangan Program di Arduino

Pada arduino dirancang program untuk menjalankan motor stepper yang terdapat pada konveyor dan wadah. Perancangan program ini mengatur stepper dikonveyor hanya untuk bergerak satu putaran setiap kali proses sedangkan untuk wadah, bergerak ke kanan atau ke kiri.

### 3.4 Pengambilan Data

Data diambil selama 5 hari dengan sample sebanyak 51 buah bawang merah. Jumlah tersebut merupakan gabungan dari 20 bawang merah kualitas 1, 11 kualitas 2 dan 20 kualitas 3.

**“Halaman ini sengaja dikosongkan”**

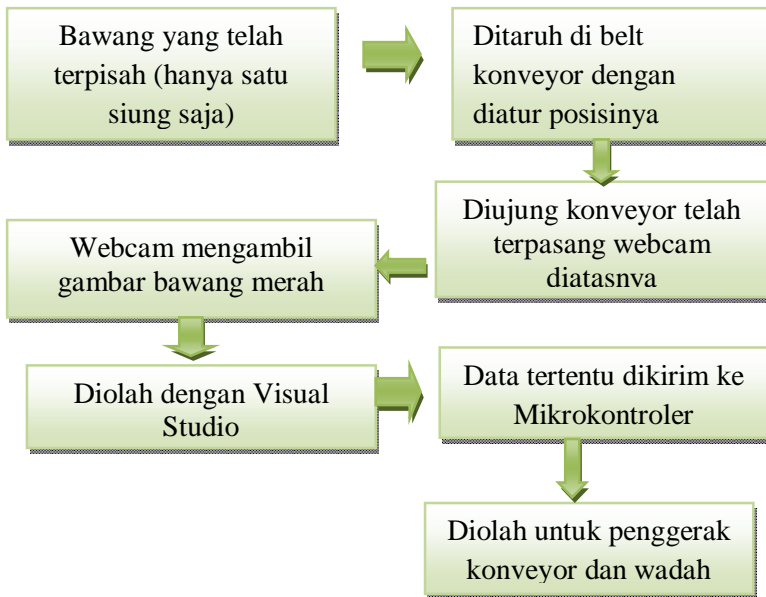


## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Gambaran Umum Alat

Prinsip dasar alat pemilah bawang merah ini adalah sebagaimana blok diagram berikut:



Gambar 4.1. Blok diagram prinsip dasar pemilah bawang merah

#### 4.2 Proses Pengolahan Citra

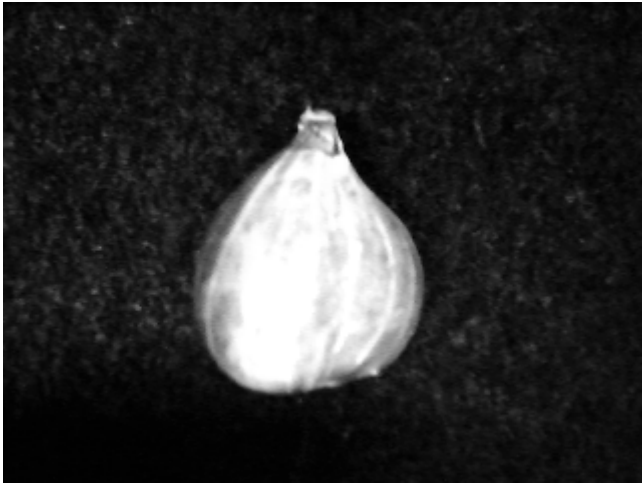
Ketika bawang merah telah berada di ujung konveyor, webcam meng-capture citra dari bawang merah. Citra tersebut termasuk citra berwarna seperti yang ditunjukkan Gambar 4.2. Citra tersebut merupakan citra RGB (Red, Green, Blue). Citra yang diambil oleh webcam tersebut berukuran  $320 \times 240 \text{ pixel}$ .

Sedangkan ukuran dari ujung-ujung citra sebenarnya 7.1 cm. Dengan begitu kita memperoleh konversi dari *pixel* ke cm, sebesar 710 mm dibagi 320 pixel adalah 2.21875 mm/pixel.



Gambar 4.2 Citra RGB Bawang Merah

Citra RGB kemudian diubah menjadi citra *Grayscale* (Gambar 4.3). Proses pengolahan citra ini bertujuan untuk menyederhanakan citra bawang merah tersebut. Secara Sederhana ini mengubah dari 3 layer (layer R, layer G dan layer B) menjadi 3 layer baru yang memiliki nilai  $R=G=B$  (seolah-olah menjadi 1 layer).



Gambar 4.3 Citra *Grayscale* Bawang Merah

Meskipun telah diubah ke citra *grayscale*(hitam-putih), citra tersebut masih perlu diolah menjadi citra biner. Citra Biner (Gambar 4.4) inilah yang diproses untuk dihitung diameter bawang merah.

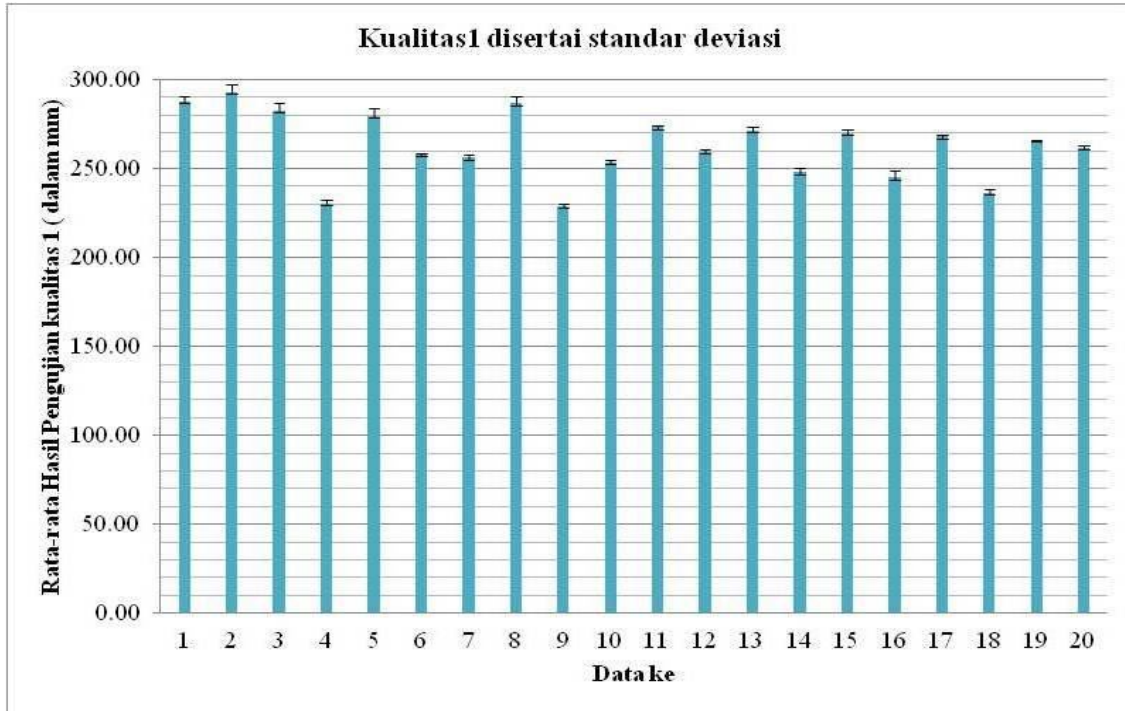


Gambar 4.4 Citra Biner Bawang Merah

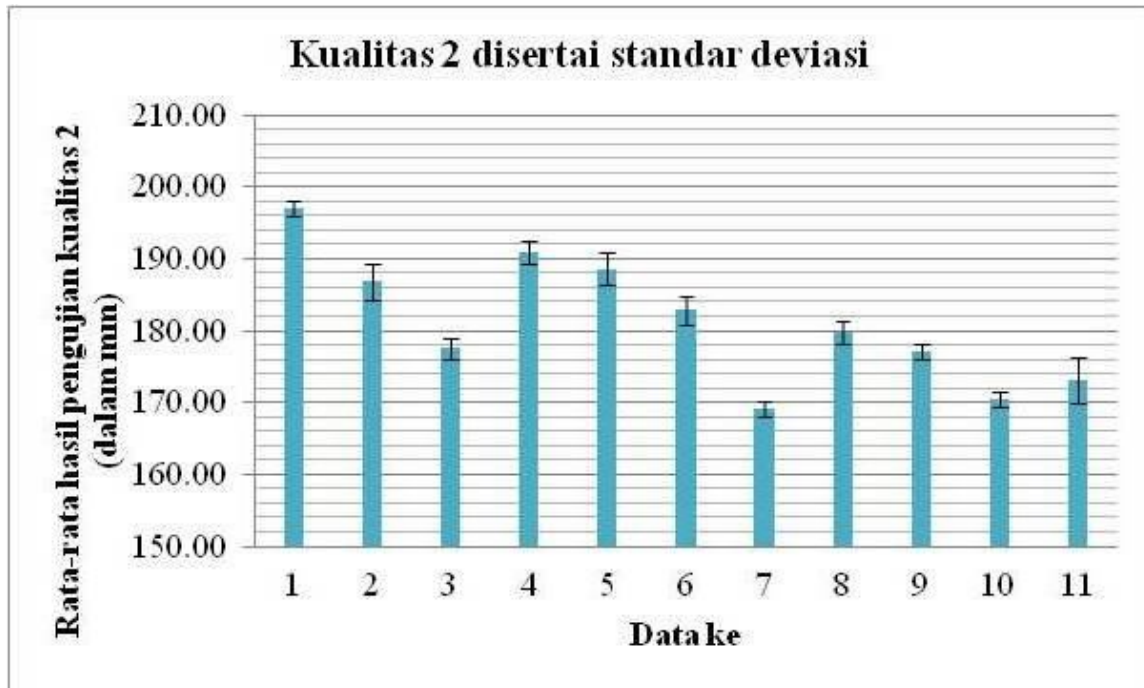
### 4.3 Data Hasil Pengujian

Data yang diambil pada pengujian ini adalah ukuran asli dan hasil pengukuran alat (ukuran dalam pixel dan ukuran dalam mm) serta kualitas. Selain itu juga diperoleh data waktu (lama) pengujian 51 siung bawang merah. Waktu yang dibutuhkan alat yang dirancang dalam Tugas Akhir ini untuk melakukan pemilahan terhadap 51 siung sampel bawang merah yang berbeda, dari pengukuran pertama sampai kelima waktunya sama, yakni selama 8 menit 30 detik.

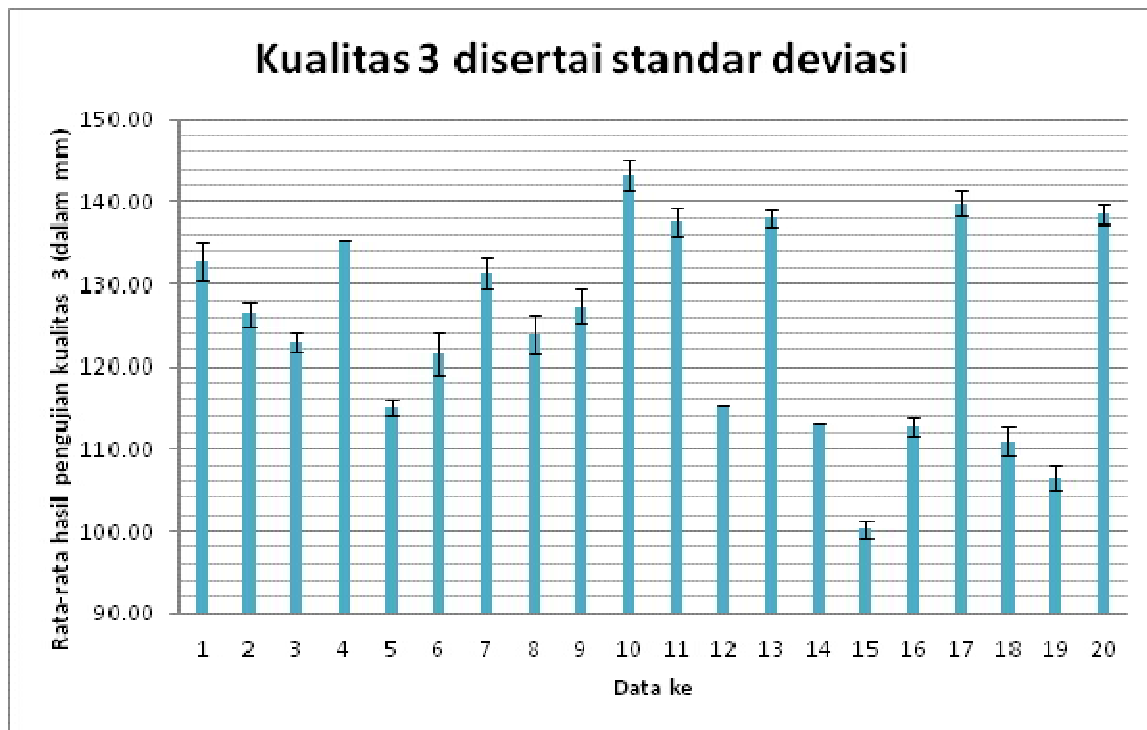
Gambar 4.5, 4.6 dan 4.7 menunjukkna hasil pengukuran kualitas bawang merah berdasarkan diameternya. Gambar 4.5 merupakan data pengujian kualitas 1 sebanyak 5 kali kemudian dirata-rata lalu dihitung standar deviasinya. Pada gambar tersebut dapat dilihat langsung bahwa 20 siung kualitas 1 ( $\text{kualitas1} > 2 \text{ cm}$ ). Gambar 4.6 merupakan grafik bar dari data pengujian kualitas 2 sebanyak 5 kali kemudian dirata-rata lalu dihitung standar deviasinya. Pada gambar tersebut dapat dilihat langsung bahwa 11 siung kualitas 2 ( $1.5\text{cm} \leq \text{kualitas 2} < 2 \text{ cm}$ ). Gambar 4.7 merupakan grafik bar dari data pengujian kualitas 3 sebanyak 5 kali kemudian dirata-rata lalu dihitung standar deviasinya. Pada gambar tersebut dapat dilihat langsung bahwa 20 siung kualitas 3 ( $\text{kualitas 3} < 1.5 \text{ cm}$ ).



Gambar 4.5 Grafik Bar Bawang Merah Kualitas 1 dengan Standar Deviasi

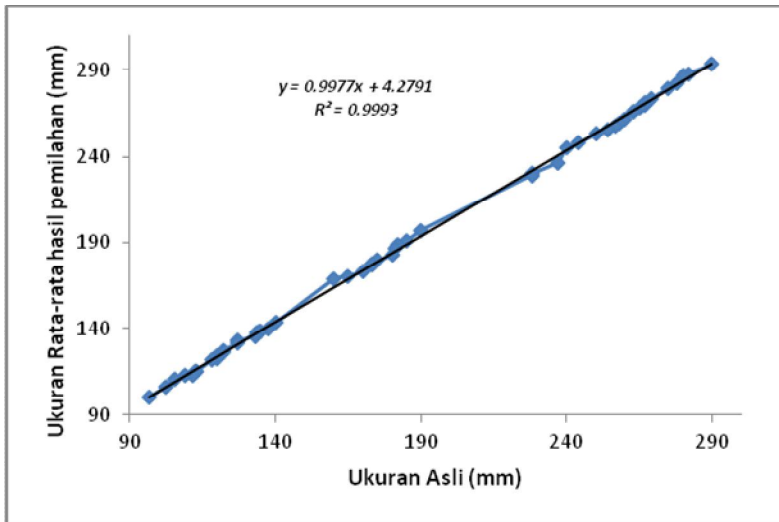


Gambar 4.6 Grafik Bar Bawang Merah Kualitas 2 disertai Standar Deviasi



Gambar 4.7 Grafik Bar Bawang Merah Kualitas 3 disertai standar Deviasi

Dari pengujian alat diperoleh nilai-nilai ukuran bawang merah yang diuji yang kemudian dibandingkan dengan nilai pengukuran manual dengan menggunakan jangka sorong. Gambar 4.8 menunjukkan perbandingan antara hasil pengukuran alat dengan hasil pengukuran manual.



Gambar 4.8 Grafik Perbandingan Ukuran Asli dan Hasil Pemilahan

Pada Gambar 4.8 dapat diketahui bahwa perbandingan antara hasil pengukuran alat dengan ukuran sebenarnya sebanding dengan persamaan  $y = 0,9977x + 4,2791$ . Selain itu juga pada sub bab sebelumnya telah diketahui nilai error relatif alat sehingga dengan demikian alat dapat dikatakan berfungsi dengan baik dalam melakukan pengukuran.



#### 4.4 Analisa Error

Dalam pengambilan data dilakukan pengulangan sebanyak 5 kali untuk tiap bawang merah yang diuji. Akibat pengulangan tersebut maka terdapat error pengukuran. Error pengukuran yang dimaksud adalah error akibat pengambilan data yang berulang. Pada pembahasan ini dibahas terkait ralat mutlak, ralat nisbi, standar deviasi serta keseksamaan pengambilan data.

Ralat mutlak merupakan kesalahan yang terjadi dalam perhitungan. Apabila nilai ralat mutlak tinggi, maka kesalahan yang dilakukan juga tinggi. Persamaan yang digunakan adalah sebagai berikut:

$$\Delta x = \sqrt{\frac{\sum (x - x_{rata})^2}{n(n-1)}} \quad (4.1)$$

Dengan  $n$  adalah jumlah pengukuran yang dilakukan. Contoh perhitungan ralat mutlak misalnya diambil data percobaan sebagaimana lampiran A maka:

$$\Delta x = \sqrt{\frac{63.01}{20}} = 1.78$$

Dari percobaan didapatkan ralat mutlak rata-rata untuk bawang merah kualitas 1 adalah sebesar 1,00; 0,75 untuk kualitas 2 ; 0,61 untuk kualitas 3. Dari data didapatkan ralat mutlak terbesar pada bawang merah kualitas 1, hal ini berarti tingkat ketidakpastian pengukurannya besar. Ralat mutlak ini memberikan range nilai hasil pengukuran yang dianggap benar adalah saat  $X = x \pm \Delta x$ . Selain itu digunakan juga ralat nisbi (relatif) dengan persamaan yang digunakan adalah

$$\text{ralat nisbi } (I) = \frac{\Delta x}{x_{rata}} \times 100\% \quad (4.2)$$

Perhitungan ralat nisbi pada bawang merah kualitas 1 (lampiran A) adalah

$$I = \frac{1.09}{121.59} \times 100\% = 0.89\%$$

Pengukuran dengan ralat nisbi terbesar adalah saat pengukuran bawang merah kualitas 3 yaitu sebesar 0.89. Hal ini berarti pada kualitas didapatkan hasil paling tidak presisi daripada pengukuran kualitas 1 dan 2. Ralat relatif berhubungan dengan ketelitian (presisi) pengukuran, semakin kecil error relatif, maka semakin besar pula tingkat ketelitian pengukuran.

Standar deviasi dan varian merupakan salah satu teknik statistik yang digunakan untuk menjelaskan homogenitas kelompok. Varians merupakan jumlah kuadrat semua deviasi nilai-nilai individual terhadap rata-rata kelompok. Sedangkan akar dari varians disebut dengan standar deviasi atau simpangan baku.

$$SD = \sqrt{\frac{\sum (x - \bar{x})^2}{(n-1)}} \quad (4.3)$$

Contoh perhitungan standar deviasi (SD) pada bawang merah kualitas 1 (Lampiran A) sebagai berikut:

$$SD = \sqrt{\frac{3.94}{4}} = 0.99$$

Standar deviasi Standar Deviasi dan Varians Simpangan baku merupakan variasi sebaran data. Semakin kecil nilai sebarannya berarti variasi nilai data makin sama. Jika sebarannya bernilai 0, maka nilai semua datanya adalah sama. Semakin besar nilai sebarannya berarti data semakin bervariasi. Sehingga standar deviasi berhubungan ke presisian data pengukuran. Dari percobaan didapatkan hasil pengukuran yang paling bervariasi

adalah saat pengukuran bawang merah kualitas 1 dengan nilai standar deviasi sebesar 4.60; dan hasil yang paling seragam adalah saat pengukuran bawang merah kualitas 3 dengan nilai standar deviasi 0.

Analisa error yang digunakan adalah ralat mutlak, ralat nisbi, standar deviasi, serta error relatif terhadap acuan. Error relatif ini merupakan error relative terhadap nilai sebenarnya, yaitu penunjukkan dengan jangka sorong. Error relatif berhubungan dengan kepresisian alat. Semakin kecil error relatif, maka sensor yang dibuat juga semakin akurat (mendekati nilai sebenarnya). Error relatif didapatkan dengan membagi mutlak kesalahan dengan nilai sebenarnya, secara matematis diberikan

$$\text{Error relatif} = \frac{|\text{alat} - \text{sebenarnya}|}{\text{sebenarnya}} \times 100\% \quad (4.4)$$

Contoh perhitungan error relatif misalnya dari data pengukuran untuk bawang merah dengan kualitas 1 diukur dengan menggunakan jangka sorong sebesar 290 mm sedangkan dari hasil pengukuran alat diperoleh nilai 293,3188 mm, maka error relatifnya

$$\begin{aligned} \text{Error relatif} &= \frac{293,3188 - 290}{290} \times 100\% = \frac{3,3188}{290} \times 100\% \\ &= 1,1444\% \end{aligned}$$

Dari percobaan didapatkan rata-rata error relatif saat pengujian bawang merah kualitas 1 dengan nilai sebesar 1.25%, kualitas 2 sebesar 3,06% dan kualitas 3 sebesar 3,04%. Hal ini berarti pengukuran saat pengujian bawang merah kualitas 1 lebih akurat daripada kualitas 2 dan 3. Hal ini dikarenakan ukurannya yang lebih besar sehingga mempermudah pengukuran. Berdasarkan analisa error ini, maka dapat disimpulkan alat yang dibuat bekerja dengan baik.

## 4.5 Pembahasan

Dalam Tugas Akhir ini dilakukan perancangan alat yang dapat memilah bawang merah berdasarkan ukurannya serta pengujian alat untuk mengetahui kecepatan alat pemilah tersebut dan membandingkannya dengan ukuran sebenarnya. Alat ini memanfaatkan pengolahan citra sebagai prinsip dasar untuk melakukan pemilahan bawang merah berdasarkan ukurannya.

Pada sub bab sebelumnya telah ditunjukkan blok diagram alat yang dibuat dalam Tugas Akhir ini. Sebagaimana yang ditunjukkan pada Gambar 4.1, cara kerja dari alat ini adalah melakukan pengambilan data dari bawang merah yang diuji kemudian mengolahnya sedemikian rupa sehingga dapat diketahui nilai ukuran bawang merah tersebut lalu melakukan pengelompokkan kualitas berdasarkan standar tertentu.

Alat ini melakukan pengambilan data berupa foto dari bawang merah yang hendak dipilah. Foto atau citra tersebut disimpan dalam format *bitmap* (BMP). Format JPG atau GIF tidak digunakan dalam alat ini karena berkas BMP umumnya tidak dimampatkan, sehingga dapat mencakup semua informasi yang ada. Berkas *bitmap* merupakan pemetaan bit. Artinya, nilai intensitas *pixel* di dalam citra dipetakan ke sejumlah bit tertentu. Peta bit yang umum adalah 8, artinya setiap *pixel* panjangnya 8 bit. Delapan bit ini merepresentasikan nilai intensitas *pixel*. Dengan demikian ada sebanyak  $2^8 = 256$  derajat keabuan, mulai dari 0 sampai 255. Seperti yang telah diketahui bahwa citra dalam format BMP ada tiga macam: citra biner, citra berwarna, dan citra hitam-putih (*grayscale*). Citra berwarna merupakan citra yang lebih umum. Warna yang terlihat pada citra *bitmap* merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru. Setiap *pixel* disusun oleh tiga komponen warna: *R* (*red*), *G* (*green*), dan *B* (*blue*). Kombinasi dari ketiga warna *RGB* tersebut menghasilkan warna yang khas untuk *pixel* yang bersangkutan. Pada citra berwarna, setiap *pixel* panjangnya 8 bit, tetapi komponen warna *RGB*-nya disimpan di dalam tabel *RGB* yang

disebut palet. Setiap komponen panjangnya 8 bit, jadi ada 256 nilai keabuan untuk warna merah, 256 nilai keabuan untuk warna hijau, dan 256 nilai keabuan untuk warna biru. Nilai setiap *pixel* tidak menyatakan derajat keabuannya secara langsung, tetapi nilai *pixel* menyatakan indeks tabel *RGB* yang memuat nilai keabuan merah (*R*), nilai keabuan hijau (*G*), dan nilai keabuan biru (*B*) untuk *pixel* yang bersangkutan. Alat yang dirancang menggunakan format BMP yang awalnya berupa citra berwarna, kemudian citra tersebut diolah menjadi citra hitam-putih. Pada citra hitam-putih, nilai  $R = G = B$  untuk menyatakan bahwa citra hitam-putih hanya mempunyai satu kanal warna. Citra hitam-putih umumnya adalah citra 8-bit. Citra yang lebih kaya warna adalah citra 24-bit. Setiap *pixel* panjangnya 24 bit, karena setiap *pixel* langsung menyatakan komponen warna merah, komponen warna hijau, dan komponen warna biru. Masing-masing komponen panjangnya 8 bit. Citra 24-bit disebut juga citra 16 juta warna, karena ia mampu menghasilkan  $2^{24} = 16.777.216$  kombinasi warna.

Meskipun telah diubah menjadi citra hitam-putih, data tersebut tidak langsung diolah untuk dilakukan pengukuran melainkan diubah kembali menjadi citra biner. Citra biner hanya mempunyai dua nilai keabuan, 0 dan 1. Oleh karena itu, 1 bit sudah cukup untuk merepresentasikan nilai *pixel*. Sehingga akan mempermudah dalam melakukan pengukuran yang kemudian akan dilanjutkan dengan pengelompokan kualitas bawang merah dari data ukuran yang diperoleh. Adapun software yang digunakan dalam pengolahan citra pada alat ini adalah Visual Studio 2010.

Dalam pengambilan data yang dilakukan ukuran yang dimaksud adalah dimensi terbesar diukur tegak lurus pada garis lurus sepanjang batang sampai akar. Sehingga dalam pengambilan data, bawang merah yang hendak diukur diatur sedemikian rupa di atas *konveyor* hingga memudahkan pengukuran.

Tiap bawang merah diambil satu data citra. Dari tiap data citra untuk satu bawang merah tersebut diperoleh lebih dari satu data yang dapat diolah sebagai ukuran bawang merah tersebut. Dari masing-masing data tersebut dilakukan sorting untuk mencari nilai paling besar yang lebih valid untuk bawang merah tersebut. Sorting dilakukan dengan menyimpan data nilai ukuran yang pertama dari bawang merah kemudian membandingkannya dengan nilai ukuran selanjutnya. Bila nilai selanjutnya lebih besar dari nilai pertama maka nilai pertama otomatis akan digantikan nilai tersebut dan dilanjutkan seterusnya sampai diperoleh nilai terbesar dari satu bawang merah dan seperti itulah seterusnya. Sorting ini perlu dilakukan dikarenakan bentuk bawang merah yang cenderung tidak teratur.

Pengelompokan kualitas bawang merah dari data ukuran yang diperoleh dilakukan dengan menggunakan standar yang telah ditentukan sebelumnya. Sistem pengelompokan berdasarkan keseragaman ukuran yang berlaku adalah : (1) kualitas 1 – diameter  $\geq 2.0$  cm, (2) kualitas 2 – diameter 1.5-2.0 cm, dan (3) kualitas 3 – diameter  $< 1.5$  cm (Adiyoga *et al.* 1998).

Dalam penelitian ini menggunakan 51 bawang merah dengan kualitas campuran. Seperti yang telah ditunjukkan pada data percobaan, dari 51 bawang merah tersebut diperoleh 20 bawang merah kualitas 1, 11 bawang merah kualitas 2 dan 20 bawang merah kualitas 3. Oleh karena itu alat ini dapat dikatakan sukses dalam pengelompokan semua bawang merah yang diujikan.

Alat ini memilah 51 bawang merah selama 8 menit 30 deti (510 detik). Waktu yang dibutuhkan tetap sama meskipun pengukuran diulangi sebanyak 5 kali. Dengan demikian dapat dikatakan bahwa alat ini memiliki kecepatan memilah 1 siung/ 10 detik bawang merah atau 6 siung/menit bawang merah.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Telah berhasil difabrikasi sebuah alat untuk memilah bawah merah berdasarkan ukurannya dengan kecepatan memilahnya sebesar 6 siung per menit. Sistem yang dibuat mampu mengenali tiga perbedaan ukuran. Dalam tugas akhir ini sistem berhasil memilah 51 bawang merah dalam tiga kategori ukuran yaitu kelas I (20 siung) dengan kesalahan 1.14%, kelas II (11 siung) dengan kesalahan 3.06% dan kelas III (20 siung) dengan kesalahan 3.04%. Perbandingan antara hasil pengukuran alat dengan ukuran sebenarnya adalah sebanding dengan persamaan  $y = 0,9977x + 4,2791$ .

#### **5.2 Saran**

Saran untuk penelitian selanjutnya dilakukan penyempurnaan terkait dengan posisi bawang merah yang tidak harus diatur posisinya.

**“Halaman ini sengaja dikosongkan”**



## LAMPIRAN A

### A.1. Error Pengukuran untuk Bawang merah kualitas 1

Cara perhitungan ralat yang digunakan:

$$\text{Ralat Mutlak}(\Delta) = \sqrt{\frac{\sum (data - data \text{ rata-rata})^2}{n(n-1)}}$$

$$\text{Ralat Nisbi (I)} = \frac{\Delta data}{data \text{ rata-rata}}$$

$$\text{Standar Deviasi} = \sqrt{\frac{\sum (data - data \text{ rata-rata})^2}{(n-1)}}$$

$$\text{Error relatif} = (\text{delta nilai} / \text{nilai sebenarnya}) \times 100\%$$

Tabel 1. Ralat Bawang merah kualitas 1

Ukuran sebenarnya	Pengukuran ke-					Rata-rata $\bar{m}$ ( $\bar{x} \times \bar{x}$ rata rata)	Ralat Mutlak	Ralat Nisbi	Standar deviasi	error relatif
	1	2	3	4	5					
290	297.31	288.44	297.31	292.88	290.66	293.32	1.78	0.61%	3.97	1.14%
282	290.66	288.44	284.00	288.44	288.44	287.99	1.09	0.38%	2.43	2.13%
280	290.66	288.44	279.56	286.22	288.44	286.66	1.91	0.67%	4.27	2.38%
278	277.34	288.44	279.56	281.78	286.22	282.67	2.06	0.73%	4.60	1.68%
275	275.13	286.22	277.34	281.78	277.34	279.56	1.98	0.71%	4.44	1.66%
269	275.13	272.91	272.91	272.91	272.91	273.35	0.44	0.16%	0.99	1.62%
267	268.47	272.91	272.91	270.69	272.91	271.58	0.89	0.33%	1.98	1.77%
267	268.47	268.47	272.91	270.69	268.47	269.80	0.89	0.33%	1.98	1.05%
265	266.25	268.47	268.47	268.47	268.47	268.03	0.44	0.17%	0.99	1.14%
263	266.25	266.25	266.25	266.25	266.25	266.25	0.00	0.00%	0.00	1.24%
260	259.59	261.81	264.03	261.81	261.81	261.81	0.70	0.27%	1.57	0.70%
258	257.38	259.59	261.81	259.59	259.59	259.59	0.70	0.27%	1.57	0.62%
257	257.38	257.38	259.59	257.38	257.38	257.82	0.44	0.17%	0.99	0.32%
254	257.38	257.38	252.94	255.16	257.38	256.04	0.89	0.35%	1.98	0.80%
250	255.16	255.16	250.72	252.94	252.94	253.38	0.83	0.33%	1.86	1.35%
244	250.72	244.06	248.50	248.50	248.50	246.06	1.09	0.44%	2.43	1.66%
240	250.72	239.63	244.06	244.06	246.28	244.95	1.80	0.74%	4.03	2.06%
237	237.41	232.97	237.41	237.41	237.41	236.52	0.89	0.38%	1.98	0.20%
228	228.53	232.97	230.75	228.53	230.75	230.31	0.83	0.36%	1.86	1.01%
228	228.53	230.75	228.53	228.53	228.53	228.98	0.44	0.19%	0.99	0.43%
Rata-rata untuk kualitas 1							1.00	0.38%	2.25	1.25%

## A.2. Error Pengukuran untuk Bawang merah kualitas 2

Tabel 2. Ralat Bawang merah kualitas 2

Ukuran sebenarnya	Pengukuran ke-				5 Rata-rata $\sum(x-x \text{ rata-rata})^2$	Ralat Mutlak	Ralat Nisbi	Standar deviasi	error relatif
	1	2	3	4					
190	197.47	197.47	197.47	195.25	197.03	3.94	0.44	0.23%	0.99
185	188.59	190.81	193.03	190.81	190.81	9.85	0.70	0.37%	1.57
182	186.38	190.81	190.81	188.59	188.59	19.69	0.99	0.53%	2.22
181	184.16	190.81	186.38	186.38	186.38	23.63	1.09	0.58%	2.43
180	181.94	186.38	181.94	181.94	182.83	15.75	0.89	0.49%	1.98
175	179.72	181.94	177.50	179.72	179.72	9.85	0.70	0.39%	1.57
173	177.50	179.72	175.28	177.50	177.50	9.85	0.70	0.40%	1.57
173	177.50	177.50	175.28	177.50	177.06	3.94	0.44	0.25%	0.99
170	168.63	177.50	173.06	173.06	173.06	39.38	1.40	0.81%	3.14
165	168.63	170.84	170.84	170.84	170.40	3.94	0.44	0.26%	0.99
160	168.63	170.84	168.63	168.63	169.07	3.94	0.44	0.26%	0.99
Rata-rata untuk kualitas 2						0.75	0.41%	1.68	3.06%

A.3. Error Pengukuran untuk Bawang merah kualitas 3

Tabel 3. Ralat Bawang merah Kualitas 3

140	146.44	142.00	144.22	142.00	142.00	142.00	143.33	15.75	0.89	0.62%	1.98	2.38%
137.5	137.56	142.00	139.78	139.78	139.78	139.78	139.78	9.85	0.70	0.50%	1.57	1.66%
135	137.56	139.78	139.78	137.56	137.56	137.56	138.45	5.91	0.54	0.39%	1.22	2.56%
134.5	137.56	139.78	137.56	137.56	137.56	137.56	138.01	3.94	0.44	0.32%	0.99	2.61%
133.5	135.34	139.78	137.56	137.56	137.56	137.56	137.56	9.85	0.70	0.51%	1.57	3.04%
133	135.34	135.34	135.34	135.34	135.34	135.34	135.34	0.00	0.00	0.00%	0.00	1.76%
127	135.34	128.69	133.13	133.13	133.13	132.68	132.68	23.63	1.09	0.82%	2.43	4.47%
127	133.13	128.69	133.13	130.91	130.91	131.35	131.35	13.78	0.83	0.63%	1.86	3.43%
122	126.47	124.25	128.69	128.69	128.69	127.36	127.36	15.75	0.89	0.70%	1.98	4.39%
122	126.47	124.25	128.69	126.47	126.47	126.47	126.47	9.85	0.70	0.55%	1.57	3.66%
120	126.47	122.03	126.47	122.03	122.03	123.81	123.81	23.63	1.09	0.88%	2.43	3.17%
120	124.25	122.03	124.25	122.03	122.03	122.92	122.92	5.91	0.54	0.44%	1.22	2.43%
118	124.25	119.81	124.25	119.81	119.81	121.59	121.59	23.63	1.09	0.89%	2.43	3.04%
113	115.38	115.38	115.38	115.38	115.38	115.38	115.38	0.00	0.00	0.00%	0.00	2.10%
112.5	115.38	113.16	115.38	115.38	115.38	114.93	114.93	3.94	0.44	0.39%	0.99	2.16%
111.5	113.16	113.16	113.16	113.16	113.16	113.16	113.16	0.00	0.00	0.00%	0.00	1.49%
109	113.16	110.94	113.16	113.16	113.16	112.71	112.71	3.94	0.44	0.39%	0.99	3.41%
105.5	110.94	108.72	113.16	110.94	110.94	110.94	110.94	9.85	0.70	0.63%	1.57	5.15%
102.5	108.72	106.50	104.28	106.50	106.50	106.50	106.50	9.85	0.70	0.66%	1.57	3.90%
96.5	102.06	99.84	99.84	99.84	99.84	100.29	100.29	3.94	0.44	0.44%	0.99	3.92%
Rata-rata untuk kualitas 3									0.61	0.49%	1.37	3.04%

## LAMPIRAN B

### B.1 *Scrip code* Visual studio 2010

#### Option Explicit On

Imports Microsoft.Office.Interop

Imports System.Runtime.InteropServices

Imports System.IO

Imports System.Drawing.Imaging

Imports System.Math

Imports System.Threading

#### Public Class Form1

Const WM\_CAP\_START = &H400S

Const WS\_CHILD = &H40000000

Const WS\_VISIBLE = &H10000000

Const WM\_CAP\_DRIVER\_CONNECT = WM\_CAP\_START  
+ 10

Const WM\_CAP\_DRIVER\_DISCONNECT =  
WM\_CAP\_START + 11

Const WM\_CAP\_EDIT\_COPY = WM\_CAP\_START + 30

Const WM\_CAP\_SEQUENCE = WM\_CAP\_START + 62

Const WM\_CAP\_FILE\_SAVEAS = WM\_CAP\_START + 23

Const WM\_CAP\_SET\_SCALE = WM\_CAP\_START + 53

Const WM\_CAP\_SET\_PREVIEWRATE =  
WM\_CAP\_START + 52

Const WM\_CAP\_SET\_PREVIEW = WM\_CAP\_START + 50

Const SWP\_NOMOVE = &H2S

Const SWP\_NOSIZE = 1

Const SWP\_NOZORDER = &H4S

Const HWND\_BOTTOM = 1

```

Declare Function capGetDriverDescriptionA Lib
"avicap32.dll" _
    (ByVal wDriverIndex As Short, _
     ByVal lpszName As String, ByVal cbName As Integer,
ByVal lpszVer As String, _
     ByVal cbVer As Integer) As Boolean

```

```

Declare Function capCreateCaptureWindowA Lib
"avicap32.dll" _
    (ByVal lpszWindowName As String, ByVal dwStyle As
Integer, _
     ByVal x As Integer, ByVal y As Integer, ByVal nWidth As
Integer, _
     ByVal nHeight As Short, ByVal hWnd As Integer, _
     ByVal nID As Integer) As Integer

```

```

Declare Function SendMessage Lib "user32" Alias
"SendMessageA" _
    (ByVal hwnd As Integer, ByVal Msg As Integer, ByVal
wParam As Integer, _
     <MarshalAs(UnmanagedType.AsAny)> ByVal lParam As
Object) As Integer

```

```

Declare Function SetWindowPos Lib "user32" Alias
"SetWindowPos" _
    (ByVal hwnd As Integer, _
     ByVal hWndInsertAfter As Integer, ByVal x As Integer,
ByVal y As Integer, _
     ByVal cx As Integer, ByVal cy As Integer, ByVal wFlags
As Integer) As Integer

```

```

Declare Function DestroyWindow Lib "user32" (ByVal hwnd
As Integer) As Boolean

```

```

Dim VideoSource As Integer
Dim hWnd As Integer
'Dim bas As Image
Dim kualitas As Integer
Dim z, ku, kual As Double
Dim myData As String
Dim count As Integer
Dim a As Integer
Dim flagRaw As Boolean = True
Private Strt As System.Threading.Thread
Dim x, y As Integer

Private Sub ListBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ListBox1.SelectedIndexChanged
    SendMessage(hWnd, WM_CAP_DRIVER_DISCONNECT,
VideoSource, 0)
    DestroyWindow(hWnd)

    VideoSource = ListBox1.SelectedIndex

    hWnd = capCreateCaptureWindowA(VideoSource,
WS_VISIBLE Or WS_CHILD, 0, 0, 0, _
    0, PictureBox1.Handle.ToInt32, 0)
    If SendMessage(hWnd, WM_CAP_DRIVER_CONNECT,
VideoSource, 0) Then

        SendMessage(hWnd, WM_CAP_SET_SCALE, True, 0)

        SendMessage(hWnd, WM_CAP_SET_PREVIEWRATE,
30, 0)

        SendMessage(hWnd, WM_CAP_SET_PREVIEW, True,
0)

```

```
SetWindowPos(hWnd, HWND_BOTTOM, 0, 0, _
    PictureBox1.Width, PictureBox1.Height, _
    SWP_NOMOVE Or SWP_NOZORDER)
```

```
Else
```

```
    DestroyWindow(hWnd)
```

```
End If
```

```
End Sub
```

```
Private Sub Form1_Load(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    ProgressBar1.Hide()
```

```
    Dim DriverName As String = Space(80)
```

```
    Dim DriverVersion As String = Space(80)
```

```
    'Dim kualitas As Integer
```

```
    count = 0
```

```
    For i As Integer = 0 To 9
```

```
        If capGetDriverDescriptionA(i, DriverName, 80,
            DriverVersion, 80) Then
```

```
            ListBox1.Items.Add(DriverName.Trim)
```

```
        End If
```

```
    Next
```

```
    Dim ports As String() = IO.Ports.SerialPort.GetPortNames()
```

```
    Dim port As String
```

```
    For Each port In ports
```

```
        ListPort.Items.Add(port)
```

```
    Next port
```

```
    kualitas = 1
```

```
    y = 0
```



End Sub

Private Sub Capture\_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles tangkap.Click

ProgressBar1.Show()

Call progress\_bar()

'SerialPort1.PortName = "COM18"

SerialPort1.PortName = ListPort.SelectedItem

SerialPort1.Parity = IO.Ports.Parity.None

SerialPort1.StopBits = IO.Ports.StopBits.One

SerialPort1.BaudRate = 9600

SerialPort1.DataBits = 8

'SerialPort1.NewLine = vbCr

Try

SerialPort1.Open()

BtnStop.Enabled = True

tangkap.Enabled = False

'BtnLoop.Enabled = True

BtnReset.Enabled = False

'LTextBox.ReadOnly = True

'ComboBox1.Enabled = False

Export.Enabled = False

SearchPort.Enabled = False

ListPort.Enabled = False

flagRaw = True

Call record()

Catch ex As Exception

MsgBox("Silahkan sambungkan perangkat dan pilih  
PORT yang digunakan.", MsgBoxStyle.Critical, "Error")

End Try

y = y + 1

```

Dim data As IDataObject
Dim bmap As Image
SendMessage(hWnd, WM_CAP_EDIT_COPY, 0, 0)

data = Clipboard.GetDataObject()
If Data.GetDataPresent(GetType(System.Drawing.Bitmap))
Then
    bmap =
CType(Data.GetData(GetType(System.Drawing.Bitmap)),
Image)
    PictureBox2.Image = bmap

    'SendMessage(hWnd,
WM_CAP_DRIVER_DISCONNECT, VideoSource, 0)
    'DestroyWindow(hWnd)
    'MsgBox("Sukses")
End If

Dim bmpNewPic As New Bitmap(PictureBox2.Image)
Dim intX As Integer 'X Counter
Dim intY As Integer 'Y Counter

Dim clrOld As Double 'Previous Color
'Dim wr, wg, wb As Double

For intX = 0 To bmpNewPic.Width - 1 'Loop Horizontally

    For intY = 0 To bmpNewPic.Height - 1 'Loop Vertically

        'Get Existing Color At Each Pixel Location and Divide
By Scale Depth
        'Any Range Between .25 and .75 Will Give optimal
Results

```

```

        clrOld = (CDBl((bmpNewPic.GetPixel(intX, intY).R) *
0.2989) + _
        (0.587) * bmpNewPic.GetPixel(intX, intY).G + _
        (bmpNewPic.GetPixel(intX, intY).B) * 0.114)

```

```

        bmpNewPic.SetPixel(intX, intY,
Color.FromArgb(clrOld, clrOld, clrOld)) 'Apply Tint
        'If (intX = 3) Then
        '    If (intY = 4) Then
        '        wr = CDBl((bmpNewPic.GetPixel(intX, intY).R)
* 0.2989)
        '        wg = CDBl((bmpNewPic.GetPixel(intX, intY).G)
* 0.587)
        '        wb = CDBl((bmpNewPic.GetPixel(intX, intY).B)
* 0.114)
        '        MsgBox("pixel 3,4 : " & wr & " , " & wg & " , "
& wb)
        '    End If
        'Else
        'End If
        Next intY
        'ProgressBar1.Value = Int(100 * intX /
(bmpNewPic.Width - 1))
        Next intX

```

```

        PictureBox2.Image = bmpNewPic 'Set New Picture
        'MsgBox("Citra : " & bmpNewPic.Height & " x " &
bmpNewPic.Width & " pixel")

```

```

Dim Pb, Pc As Integer
Dim Rt, vM, vH, vB As Double
Dim gamrbiner As New Bitmap(PictureBox2.Image)
'ProgressBar1.Show()

```

```

For Pb = 0 To gambrbinner.Height - 1
  For Pc = 0 To gambrbinner.Width - 1
    vM = CDBl(gambrbinner.GetPixel(Pc, Pb).R) + 150
    vH = CDBl(gambrbinner.GetPixel(Pc, Pb).G) + 150
    vB = CDBl(gambrbinner.GetPixel(Pc, Pb).B) + 150
    Rt = (vM + vH + vB) / 3

    If (Rt >= 255) Then
      vM = 255
    Else : vM = 0
    End If

    If (vB >= 255) Then
      vB = 255
    Else : vB = 0
    End If

    If (vH >= 255) Then
      vH = 255
    Else : vH = 0
    End If

    gambrbinner.SetPixel(Pc, Pb, Color.FromArgb(vM, vH,
vB))
  'End If
Next

Next

PictureBox2.Image = gambrbinner

'ini proses hitungnya
Dim Gb, Gc, kirim As Integer
Dim GM As Double
'Dim gamabrbiner As New Bitmap(PictureBox2.Image)
'ProgressBar1.Show()

```

```

ku = 0
For Gb = 0 To gambrbiner.Height - 1
    z = 0
    For Gc = 0 To gambrbiner.Width - 1
        GM = CDBl(gambrbiner.GetPixel(Gc, Gb).R)
        If (GM = 255) Then
            z = z + 1
        Else
        End If
    Next

    If (ku < z) Then
        ku = z
    End If
    'ProgressBar1.Value = Int(100 * Gb / (gambrbiner.Height
- 1))

Next

kual = ku * 2.21875
kirim = kualitas
If (kual >= 200) Then
    kualitas = 1
Else
    If (kual < 150) Then
        kualitas = 3
    Else
        kualitas = 2
    End If
End If
'MsgBox("Ukuran Bawang Merah : " & kual & " mm
sehingga termasuk kualitas : " & kualitas & "")

Try

```

```

    If ((kualitas - kirim) = 0) Then
        SerialPort1.Write("a") 'diam
        Threading.Thread.Sleep(500)
    Else
        If ((kualitas - kirim) = 1) Then
            SerialPort1.Write("b") 'kanan1
            Threading.Thread.Sleep(500)
        Else
            If ((kualitas - kirim) = 2) Then
                SerialPort1.Write("c") 'kanan2
                Threading.Thread.Sleep(500)
            Else
                If ((kualitas - kirim) = -1) Then
                    SerialPort1.Write("d") 'kiri1
                    Threading.Thread.Sleep(500)
                Else
                    If ((kualitas - kirim) = -2) Then
                        SerialPort1.Write("e") 'kiri2
                        Threading.Thread.Sleep(500)
                    Else
                        MsgBox("Maaf Ada Sedikit Masalah",
MsgBoxStyle.Critical, "Error")
                    End If
                End If
            End If
        End If
    End If
Catch ex As Exception

End Try

count = count + 1
a = count - 1
If flagRaw = True Then
    AccesControll()

```

```

        gridData.Rows.GetNextRow(count,
DataGridViewElementStates.ReadOnly)

```

```

End If

```

```

    gridData.Item(0, a).Value = count

```

```

    AccesControl2()

```

```

    Timer1.Enabled = True

```

```

If y = 60 Then

```

```

    BtnStop.PerformClick()

```

```

End If

```

```

End Sub

```

```

Private Sub record()

```

```

    If flagRaw = True Then

```

```

        gridData.Rows.GetNextRow(count,

```

```

DataGridViewElementStates.ReadOnly)

```

```

    End If

```

```

End Sub

```

```

Private Sub AccesControl1()

```

```

    If Me.InvokeRequired Then

```

```

        Me.Invoke(New MethodInvoker(AddressOf
AccesControl1))

```

```

    Else

```

```

        gridData.Rows.Add()

```

```

    End If

```

```

End Sub

```

```

Private Sub AccesControl2()

```

```

    'Call function_split()

```

```

    'If myDataSplit(3) > 1 Then

```

```

'count = count + 1
a = count - 1
If Me.InvokeRequired Then
    Me.Invoke(New MethodInvoker(AddressOf
AccesControl2))

    ' MsgBox("Silahkan bisa", MsgBoxStyle.Exclamation,
"Peringatan")
Else
    gridData.Item(0, a).Value = count
    gridData.Item(1, a).Value = ku
    gridData.Item(2, a).Value = kual
    gridData.Item(3, a).Value = kualitas

End If

End Sub

Private Sub Export_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Export.Click
    Call progress_bar()
    Dim appsExcel As New Excel.Application
    Dim excelSheet, excelBook As Object

    Dim i, j As Integer

    Dim oldCi As System.Globalization.CultureInfo =
System.Threading.Thread.CurrentThread.CurrentCulture
    System.Threading.Thread.CurrentThread.CurrentCulture =
New System.Globalization.CultureInfo("en-US")
    excelBook = appsExcel.Workbooks.Add
    System.Threading.Thread.CurrentThread.CurrentCulture =
oldCi
    excelSheet = appsExcel.Worksheets(1)
'Dim oRng As Excel.Range

```



```
appsExcel = CreateObject("Excel.Application")
```

```
'appsExcel.Visible = True
```

```
With excelSheet.Range("A1", "D1")
```

```
.Font.Bold = True
```

```
.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter
```

```
End With
```

```
With excelSheet
```

```
.Range("A1:D1").Font.Bold = True
```

```
.Cells(1, 1) = "No"
```

```
.Cells(1, 2) = "Pixel"
```

```
.cells(1, 3) = "Ukuran"
```

```
.cells(1, 4) = "Kualitas"
```

```
End With
```

```
For i = 2 To gridData.RowCount
```

```
    excelSheet.cells(i, 1) = gridData.Rows(i -
```

```
2).Cells("Nomor").Value
```

```
    For j = 1 To gridData.ColumnCount - 1
```

```
        excelSheet.cells(i, j + 1) = gridData.Rows(i -
```

```
2).Cells(j).Value
```

```
    Next
```

```
Next
```

```
excelBook.SaveAs("C:\Users\hidayat\Desktop\example.xlsx",
```

```
Excel.XlFileFormat.xlWorkbookDefault, Type.Missing,
Type.Missing, Type.Missing, Type.Missing, _
    Excel.XlSaveAsAccessMode.xlNoChange,
Excel.XlSaveConflictResolution.xlLocalSessionChanges)
```

```
appsExcel.Visible = True
excelSheet = Nothing
excelBook = Nothing
'appsExcel = Nothing
MsgBox("Hasil dari export ke excel bisa" + vbNewLine + _
    "di lihat di C:\Users\hidayat\Desktop\example.xlsx",
MsgBoxStyle.Information, "export to excel")
End Sub
```

```
Private Sub BtnStop_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BtnStop.Click
    Call progress_bar()
    Try
        SerialPort1.Close()
        tangkap.Enabled = True
        BtnStop.Enabled = False
        Export.Enabled = True
        'LTextBox.ReadOnly = False
        'ComboBox1.Enabled = True
        BtnReset.Enabled = True
        SearchPort.Enabled = True
        ListPort.Enabled = True
        flagRaw = False
        Timer1.Enabled = False
    Catch ex As Exception
        MsgBox("Failed to close port.", MsgBoxStyle.Critical,
"Error")
    End Try
End Sub
```

```

Private Sub SearchPort_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles SearchPort.Click
    Call progress_bar()
    Dim ports As String() = IO.Ports.SerialPort.GetPortNames()
    Dim port As String

    For Each port In ports
        ListPort.Items.Add(port)
    Next port
End Sub

'Public Sub jeda(ByVal detik As Integer)
'    Dim awal As Date = Now()
'    While (DateDiff("s", awal, Now()) <= detik)
'        'do nothing
'    End While 'diganti Wend jika pakai VB6
'End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Timer1.Tick
    SerialPort1.Close()
    tangkap.Enabled = True
    tangkap.PerformClick()
End Sub

Private Sub BtnReset_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BtnReset.Click
    Call progress_bar()
    gridData.Rows.Clear()
    count = 0
    ku = 0
    kual = 0
    kualitas = 1

    flagRaw = True

```

```

    TextBox1.Text = 0
    Export.Enabled = False
End Sub

```

```

Private Sub progress_bar()
    ProgressBar1.Minimum = 0
    ProgressBar1.Maximum = 100
    Dim i As Integer
    For i = 0 To 100
        ProgressBar1.Value = i
        Application.DoEvents()
    Next
End Sub

```

```

Private Sub btnAbout_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnAbout.Click
    Form2.Visible = True
    Timer2.Enabled = True

```

```

End Sub

```

```

Private Sub BtnHelp_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BtnHelp.Click

```

```

    MsgBox("Silahkan klik list Box dan pilih webcam" +
vbNewLine + _
        "Silahkan Search port dan pilih Port (biasanya COM18)
" + vbNewLine + _
        "Klik tombol START untuk memulai proses pemilahan
bawang merah" + vbNewLine + vbNewLine + "PERHATIAN!"
+ vbNewLine + _
        "Silahkan Melanjutkan", MsgBoxStyle.Information,
"Help")

```

```

End Sub

```

```
Private Sub Timer2_Tick(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Timer2.Tick  
    Form2.Visible = False  
    'Timer2.Enabled = False  
End Sub
```

```
End Class
```

## B.2 Scrip code Arduino

```
#include <Stepper.h>

const int step_360 = 200; // 360 number of steps per/revolution
(satu putaran)
// initialize the stepper library on pins 2-5 n 8-11
Stepper myStepper2(step_360,8,9,10,11);
Stepper myStepper1(step_360,2,3,4,5);

String message,text;
int led = 13;
void setup()
{
  // set the speed at 60 rpm:
  myStepper1.setSpeed(60); //stepper 1 for wadah
  myStepper2.setSpeed(60); //stepper 2 for konveyor

  // initialize the serial port:
  //pinMode(31, OUTPUT); //pin 31 untuk stepper 1
  //pinMode(35, OUTPUT); //pin 35 untuk stepper 2
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);

  Serial.begin(9600);
}
void loop()
{
  //digitalWrite(31, LOW);
```

```

digitalWrite(35, LOW);
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);
digitalWrite(5, HIGH);
digitalWrite(8, HIGH);
digitalWrite(9, HIGH);
digitalWrite(10, HIGH);
digitalWrite(11, HIGH);
String text = "";
if (Serial.available()>0)
{digitalWrite(31, HIGH); // turn the pin 31 on (HIGH is the
voltage level)
//Serial.println("serial bisa");
while(Serial.available()>0)
{
    text += char(Serial.read());
    delay(10);
}
message = text;
//Serial.println(message);
if (message=="a")
    {//Serial.println(message);
    }
else
{

    if (message=="b")//kanan 1 langkah

        { myStepper1.step(250);

        //Serial.println(message);
        }

    else

```

```

{
  if (message=="d")//kiri 1 langkah

      { myStepper1.step(-250);

        //Serial.println(message);
      }
  else
  {
    if (message=="c")//ke kanan 2 langkah

        { myStepper1.step(500);

          Serial.println(message);
        }
    else
    {
      if (message=="e")//kekiri 2 langkah

          { myStepper1.step(-500);

            Serial.println(message);
          }
    }
  }
}

digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);
digitalWrite(5, HIGH);
//digitalWrite(31, LOW);
delay(500);
//digitalWrite(35, HIGH);

```



```
    myStepper2.step(-200);

    // delay(1000);
    // digitalWrite(35, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(9, HIGH);
    digitalWrite(10, HIGH);
    digitalWrite(11, HIGH);
    //Serial.println("1");
  }

}
```

**“Halaman ini sengaja dikosongkan”**

## LAMPIRAN C

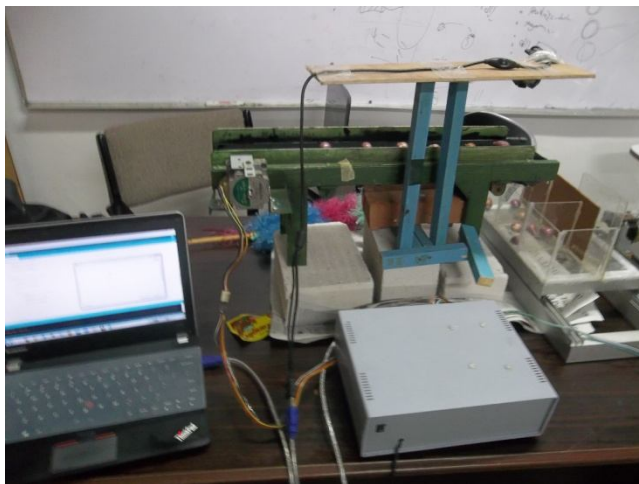
### Dokumentasi Penelitian



Gambar 1. Proses pembuatan alat



Gambar 2. Posisi Bawang Merah Pada konveyor



Gambar 3. Tampilan alat



Gambar 4. Hasil Pemilahan

## BIODATA PENULIS



Penulis bernama **Moh. Roni Hidayat** merupakan anak pertama dari tiga bersaudara dengan Bapak bernama Umar Fauzi dan ibu bernama Sarinah. Ia dilahirkan di Kediri pada tanggal 26 Agustus 1991. Penulis memiliki *background* pendidikan formal di Madrasah Ibtidaiyah Salafiyah Banaran-

Tunglur Badas Kediri (2003), MTsN Model 1 Pare Kediri, serta jenjang SLTA di SMAN 2 Pare Kediri. Setelah lulus SLTA, penulis melanjutkan kuliah di Jurusan Fisika Institut Teknologi Sepuluh Nopember Surabaya (2009 – sekarang).

Orang tua penulis adalah petani bawang merah, sehingga tugas akhir ini berkaitan dengan bawang merah. Selain itu penulis memiliki hobi belajar IT (meskipun hanya sekedar membaca), karena itu tugas akhir ini sedikit mengenai pemrograman.

*“Halaman ini sengaja dikosongkan”*